

optimizer

SGD

```
In [15]: import sys, os
sys.path.append(os.pardir) # 親ディレクトリのファイルをインポートするための設定
import numpy as np
from collections import OrderedDict
from common import layers
from data.mnist import load_mnist
import matplotlib.pyplot as plt
from multi_layer_net import MultiLayerNet

# データの読み込み
(x_train, d_train), (x_test, d_test) = load_mnist(normalize=True, one_hot_label=True)

print("データ読み込み完了")

# batch_normalizationの設定 =====
#use_batchnorm = True
use_batchnorm = False
# =====

network = MultiLayerNet(input_size=784, hidden_size_list=[40, 20], output_size=10, activation='sigmoid', weight_init_std=0.01,
                        use_batchnorm=use_batchnorm)

iters_num = 1000
train_size = x_train.shape[0]
batch_size = 100
learning_rate = 0.01

train_loss_list = []
accuracies_train = []
accuracies_test = []

plot_interval=10

for i in range(iters_num):
    idx_choice = np.random.choice(train_size, batch_size)
    x_batch = x_train[idx_choice]
    d_batch = d_train[idx_choice]

    # 勾配
```

```
grad = network.gradient(x_batch, d_batch)

for key in ('W1', 'W2', 'W3', 'b1', 'b2', 'b3'):
    network.params[key] -= learning_rate * grad[key]

    loss = network.loss(x_batch, d_batch)
    train_loss_list.append(loss)

if (i + 1) % plot_interval == 0:
    accr_test = network.accuracy(x_test, d_test)
    accuracies_test.append(accr_test)
    accr_train = network.accuracy(x_batch, d_batch)
    accuracies_train.append(accr_train)

    print('Generation: ' + str(i+1) + '. 正答率(トレーニング) = ' + str(accr_train))
    print('                : ' + str(i+1) + '. 正答率(テスト) = ' + str(accr_test))

lists = range(0, iters_num, plot_interval)
plt.plot(lists, accuracies_train, label="training set")
plt.plot(lists, accuracies_test, label="test set")
plt.legend(loc="lower right")
plt.title("accuracy")
plt.xlabel("count")
plt.ylabel("accuracy")
plt.ylim(0, 1.0)
# グラフの表示
plt.show()
```

データ読み込み完了

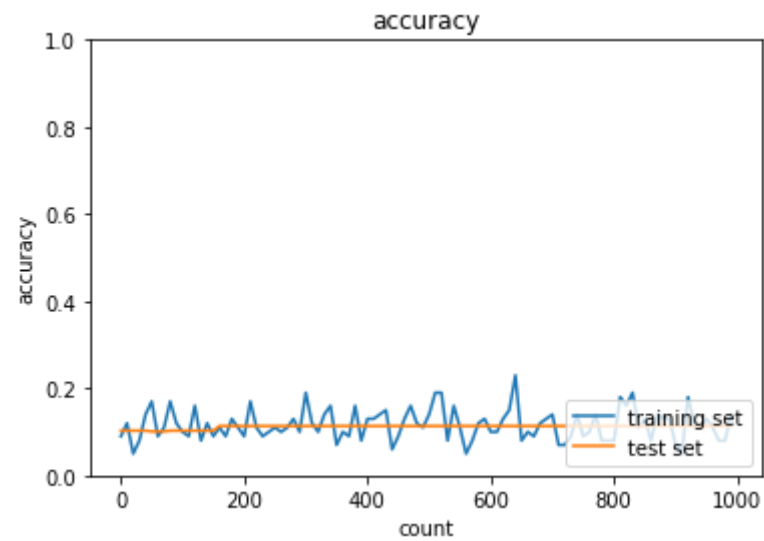
Generation: 10. 正答率(トレーニング) = 0.09
: 10. 正答率(テスト) = 0.1028
Generation: 20. 正答率(トレーニング) = 0.12
: 20. 正答率(テスト) = 0.1028
Generation: 30. 正答率(トレーニング) = 0.05
: 30. 正答率(テスト) = 0.1028
Generation: 40. 正答率(トレーニング) = 0.08
: 40. 正答率(テスト) = 0.1028
Generation: 50. 正答率(トレーニング) = 0.14
: 50. 正答率(テスト) = 0.1028
Generation: 60. 正答率(トレーニング) = 0.17
: 60. 正答率(テスト) = 0.101
Generation: 70. 正答率(トレーニング) = 0.09
: 70. 正答率(テスト) = 0.101
Generation: 80. 正答率(トレーニング) = 0.11
: 80. 正答率(テスト) = 0.101
Generation: 90. 正答率(トレーニング) = 0.17
: 90. 正答率(テスト) = 0.1028
Generation: 100. 正答率(トレーニング) = 0.12
: 100. 正答率(テスト) = 0.1028
Generation: 110. 正答率(トレーニング) = 0.1
: 110. 正答率(テスト) = 0.1028
Generation: 120. 正答率(トレーニング) = 0.09
: 120. 正答率(テスト) = 0.1028
Generation: 130. 正答率(トレーニング) = 0.16
: 130. 正答率(テスト) = 0.1028
Generation: 140. 正答率(トレーニング) = 0.08
: 140. 正答率(テスト) = 0.1028
Generation: 150. 正答率(トレーニング) = 0.12
: 150. 正答率(テスト) = 0.1028
Generation: 160. 正答率(トレーニング) = 0.09
: 160. 正答率(テスト) = 0.1028
Generation: 170. 正答率(トレーニング) = 0.11
: 170. 正答率(テスト) = 0.1135
Generation: 180. 正答率(トレーニング) = 0.09
: 180. 正答率(テスト) = 0.1135
Generation: 190. 正答率(トレーニング) = 0.13
: 190. 正答率(テスト) = 0.1135
Generation: 200. 正答率(トレーニング) = 0.11
: 200. 正答率(テスト) = 0.1135

Generation: 210. 正答率(トレーニング) = 0.09
: 210. 正答率(テスト) = 0.1135
Generation: 220. 正答率(トレーニング) = 0.17
: 220. 正答率(テスト) = 0.1135
Generation: 230. 正答率(トレーニング) = 0.11
: 230. 正答率(テスト) = 0.1135
Generation: 240. 正答率(トレーニング) = 0.09
: 240. 正答率(テスト) = 0.1135
Generation: 250. 正答率(トレーニング) = 0.1
: 250. 正答率(テスト) = 0.1135
Generation: 260. 正答率(トレーニング) = 0.11
: 260. 正答率(テスト) = 0.1135
Generation: 270. 正答率(トレーニング) = 0.1
: 270. 正答率(テスト) = 0.1135
Generation: 280. 正答率(トレーニング) = 0.11
: 280. 正答率(テスト) = 0.1135
Generation: 290. 正答率(トレーニング) = 0.13
: 290. 正答率(テスト) = 0.1135
Generation: 300. 正答率(トレーニング) = 0.1
: 300. 正答率(テスト) = 0.1135
Generation: 310. 正答率(トレーニング) = 0.19
: 310. 正答率(テスト) = 0.1135
Generation: 320. 正答率(トレーニング) = 0.12
: 320. 正答率(テスト) = 0.1135
Generation: 330. 正答率(トレーニング) = 0.1
: 330. 正答率(テスト) = 0.1135
Generation: 340. 正答率(トレーニング) = 0.14
: 340. 正答率(テスト) = 0.1135
Generation: 350. 正答率(トレーニング) = 0.16
: 350. 正答率(テスト) = 0.1135
Generation: 360. 正答率(トレーニング) = 0.07
: 360. 正答率(テスト) = 0.1135
Generation: 370. 正答率(トレーニング) = 0.1
: 370. 正答率(テスト) = 0.1135
Generation: 380. 正答率(トレーニング) = 0.09
: 380. 正答率(テスト) = 0.1135
Generation: 390. 正答率(トレーニング) = 0.16
: 390. 正答率(テスト) = 0.1135
Generation: 400. 正答率(トレーニング) = 0.08
: 400. 正答率(テスト) = 0.1135
Generation: 410. 正答率(トレーニング) = 0.13
: 410. 正答率(テスト) = 0.1135

Generation: 420. 正答率(トレーニング) = 0.13
: 420. 正答率(テスト) = 0.1135
Generation: 430. 正答率(トレーニング) = 0.14
: 430. 正答率(テスト) = 0.1135
Generation: 440. 正答率(トレーニング) = 0.15
: 440. 正答率(テスト) = 0.1135
Generation: 450. 正答率(トレーニング) = 0.06
: 450. 正答率(テスト) = 0.1135
Generation: 460. 正答率(トレーニング) = 0.09
: 460. 正答率(テスト) = 0.1135
Generation: 470. 正答率(トレーニング) = 0.13
: 470. 正答率(テスト) = 0.1135
Generation: 480. 正答率(トレーニング) = 0.16
: 480. 正答率(テスト) = 0.1135
Generation: 490. 正答率(トレーニング) = 0.12
: 490. 正答率(テスト) = 0.1135
Generation: 500. 正答率(トレーニング) = 0.11
: 500. 正答率(テスト) = 0.1135
Generation: 510. 正答率(トレーニング) = 0.14
: 510. 正答率(テスト) = 0.1135
Generation: 520. 正答率(トレーニング) = 0.19
: 520. 正答率(テスト) = 0.1135
Generation: 530. 正答率(トレーニング) = 0.19
: 530. 正答率(テスト) = 0.1135
Generation: 540. 正答率(トレーニング) = 0.08
: 540. 正答率(テスト) = 0.1135
Generation: 550. 正答率(トレーニング) = 0.16
: 550. 正答率(テスト) = 0.1135
Generation: 560. 正答率(トレーニング) = 0.11
: 560. 正答率(テスト) = 0.1135
Generation: 570. 正答率(トレーニング) = 0.05
: 570. 正答率(テスト) = 0.1135
Generation: 580. 正答率(トレーニング) = 0.08
: 580. 正答率(テスト) = 0.1135
Generation: 590. 正答率(トレーニング) = 0.12
: 590. 正答率(テスト) = 0.1135
Generation: 600. 正答率(トレーニング) = 0.13
: 600. 正答率(テスト) = 0.1135
Generation: 610. 正答率(トレーニング) = 0.1
: 610. 正答率(テスト) = 0.1135
Generation: 620. 正答率(トレーニング) = 0.1
: 620. 正答率(テスト) = 0.1135

Generation: 630. 正答率(トレーニング) = 0.13
: 630. 正答率(テスト) = 0.1135
Generation: 640. 正答率(トレーニング) = 0.15
: 640. 正答率(テスト) = 0.1135
Generation: 650. 正答率(トレーニング) = 0.23
: 650. 正答率(テスト) = 0.1135
Generation: 660. 正答率(トレーニング) = 0.08
: 660. 正答率(テスト) = 0.1135
Generation: 670. 正答率(トレーニング) = 0.1
: 670. 正答率(テスト) = 0.1135
Generation: 680. 正答率(トレーニング) = 0.09
: 680. 正答率(テスト) = 0.1135
Generation: 690. 正答率(トレーニング) = 0.12
: 690. 正答率(テスト) = 0.1135
Generation: 700. 正答率(トレーニング) = 0.13
: 700. 正答率(テスト) = 0.1135
Generation: 710. 正答率(トレーニング) = 0.14
: 710. 正答率(テスト) = 0.1135
Generation: 720. 正答率(トレーニング) = 0.07
: 720. 正答率(テスト) = 0.1135
Generation: 730. 正答率(トレーニング) = 0.07
: 730. 正答率(テスト) = 0.1135
Generation: 740. 正答率(トレーニング) = 0.09
: 740. 正答率(テスト) = 0.1135
Generation: 750. 正答率(トレーニング) = 0.14
: 750. 正答率(テスト) = 0.1135
Generation: 760. 正答率(トレーニング) = 0.09
: 760. 正答率(テスト) = 0.1135
Generation: 770. 正答率(トレーニング) = 0.1
: 770. 正答率(テスト) = 0.1135
Generation: 780. 正答率(トレーニング) = 0.14
: 780. 正答率(テスト) = 0.1135
Generation: 790. 正答率(トレーニング) = 0.08
: 790. 正答率(テスト) = 0.1135
Generation: 800. 正答率(トレーニング) = 0.08
: 800. 正答率(テスト) = 0.1135
Generation: 810. 正答率(トレーニング) = 0.08
: 810. 正答率(テスト) = 0.1135
Generation: 820. 正答率(トレーニング) = 0.18
: 820. 正答率(テスト) = 0.1135
Generation: 830. 正答率(トレーニング) = 0.16
: 830. 正答率(テスト) = 0.1135

Generation: 840. 正答率(トレーニング) = 0.19
: 840. 正答率(テスト) = 0.1135
Generation: 850. 正答率(トレーニング) = 0.11
: 850. 正答率(テスト) = 0.1135
Generation: 860. 正答率(トレーニング) = 0.13
: 860. 正答率(テスト) = 0.1135
Generation: 870. 正答率(トレーニング) = 0.08
: 870. 正答率(テスト) = 0.1135
Generation: 880. 正答率(トレーニング) = 0.14
: 880. 正答率(テスト) = 0.1135
Generation: 890. 正答率(トレーニング) = 0.13
: 890. 正答率(テスト) = 0.1135
Generation: 900. 正答率(トレーニング) = 0.13
: 900. 正答率(テスト) = 0.1135
Generation: 910. 正答率(トレーニング) = 0.07
: 910. 正答率(テスト) = 0.1135
Generation: 920. 正答率(トレーニング) = 0.05
: 920. 正答率(テスト) = 0.1135
Generation: 930. 正答率(トレーニング) = 0.18
: 930. 正答率(テスト) = 0.1135
Generation: 940. 正答率(トレーニング) = 0.12
: 940. 正答率(テスト) = 0.1135
Generation: 950. 正答率(トレーニング) = 0.11
: 950. 正答率(テスト) = 0.1135
Generation: 960. 正答率(トレーニング) = 0.13
: 960. 正答率(テスト) = 0.1135
Generation: 970. 正答率(トレーニング) = 0.11
: 970. 正答率(テスト) = 0.1135
Generation: 980. 正答率(トレーニング) = 0.08
: 980. 正答率(テスト) = 0.1135
Generation: 990. 正答率(トレーニング) = 0.08
: 990. 正答率(テスト) = 0.1135
Generation: 1000. 正答率(トレーニング) = 0.14
: 1000. 正答率(テスト) = 0.1135



[try] 学習率を変えてみよう

SGD

```
In [18]: import sys, os
sys.path.append(os.pardir) # 親ディレクトリのファイルをインポートするための設定
import numpy as np
from collections import OrderedDict
from common import layers
from data.mnist import load_mnist
import matplotlib.pyplot as plt
from multi_layer_net import MultiLayerNet

# データの読み込み
(x_train, d_train), (x_test, d_test) = load_mnist(normalize=True, one_hot_label=True)

print("データ読み込み完了")

# batch_normalizationの設定 =====
#use_batchnorm = True
use_batchnorm = False
# =====

network = MultiLayerNet(input_size=784, hidden_size_list=[40, 20], output_size=10, activation='sigmoid', weight_init_std=0.01,
                        use_batchnorm=use_batchnorm)

iters_num = 1000
train_size = x_train.shape[0]
batch_size = 100
learning_rate = 1

train_loss_list = []
accuracies_train = []
accuracies_test = []

plot_interval=10

for i in range(iters_num):
    idx_choice = np.random.choice(train_size, batch_size)
    x_batch = x_train[idx_choice]
    d_batch = d_train[idx_choice]

    # 勾配
```

```
grad = network.gradient(x_batch, d_batch)

for key in ('W1', 'W2', 'W3', 'b1', 'b2', 'b3'):
    network.params[key] -= learning_rate * grad[key]

    loss = network.loss(x_batch, d_batch)
    train_loss_list.append(loss)

if (i + 1) % plot_interval == 0:
    accr_test = network.accuracy(x_test, d_test)
    accuracies_test.append(accr_test)
    accr_train = network.accuracy(x_batch, d_batch)
    accuracies_train.append(accr_train)

    print('Generation: ' + str(i+1) + '. 正答率(トレーニング) = ' + str(accr_train))
    print('                : ' + str(i+1) + '. 正答率(テスト) = ' + str(accr_test))

lists = range(0, iters_num, plot_interval)
plt.plot(lists, accuracies_train, label="training set")
plt.plot(lists, accuracies_test, label="test set")
plt.legend(loc="lower right")
plt.title("accuracy")
plt.xlabel("count")
plt.ylabel("accuracy")
plt.ylim(0, 1.0)
# グラフの表示
plt.show()
```

データ読み込み完了

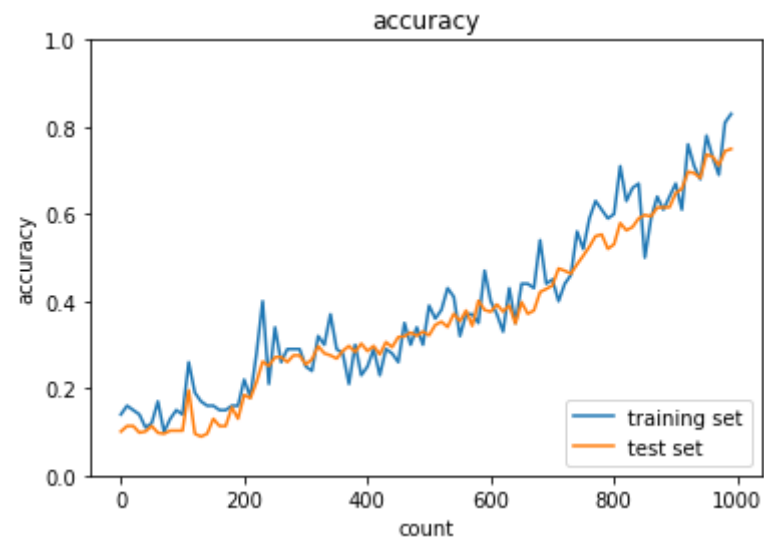
Generation: 10. 正答率(トレーニング) = 0.14
: 10. 正答率(テスト) = 0.1009
Generation: 20. 正答率(トレーニング) = 0.16
: 20. 正答率(テスト) = 0.1135
Generation: 30. 正答率(トレーニング) = 0.15
: 30. 正答率(テスト) = 0.1135
Generation: 40. 正答率(トレーニング) = 0.14
: 40. 正答率(テスト) = 0.0982
Generation: 50. 正答率(トレーニング) = 0.11
: 50. 正答率(テスト) = 0.101
Generation: 60. 正答率(トレーニング) = 0.12
: 60. 正答率(テスト) = 0.1135
Generation: 70. 正答率(トレーニング) = 0.17
: 70. 正答率(テスト) = 0.098
Generation: 80. 正答率(トレーニング) = 0.1
: 80. 正答率(テスト) = 0.0958
Generation: 90. 正答率(トレーニング) = 0.13
: 90. 正答率(テスト) = 0.1028
Generation: 100. 正答率(トレーニング) = 0.15
: 100. 正答率(テスト) = 0.1028
Generation: 110. 正答率(トレーニング) = 0.14
: 110. 正答率(テスト) = 0.1028
Generation: 120. 正答率(トレーニング) = 0.26
: 120. 正答率(テスト) = 0.1958
Generation: 130. 正答率(トレーニング) = 0.19
: 130. 正答率(テスト) = 0.0958
Generation: 140. 正答率(トレーニング) = 0.17
: 140. 正答率(テスト) = 0.0892
Generation: 150. 正答率(トレーニング) = 0.16
: 150. 正答率(テスト) = 0.0958
Generation: 160. 正答率(トレーニング) = 0.16
: 160. 正答率(テスト) = 0.1301
Generation: 170. 正答率(トレーニング) = 0.15
: 170. 正答率(テスト) = 0.1135
Generation: 180. 正答率(トレーニング) = 0.15
: 180. 正答率(テスト) = 0.1135
Generation: 190. 正答率(トレーニング) = 0.16
: 190. 正答率(テスト) = 0.1563
Generation: 200. 正答率(トレーニング) = 0.16
: 200. 正答率(テスト) = 0.1303

Generation: 210. 正答率(トレーニング) = 0.22
: 210. 正答率(テスト) = 0.1848
Generation: 220. 正答率(トレーニング) = 0.18
: 220. 正答率(テスト) = 0.1773
Generation: 230. 正答率(トレーニング) = 0.28
: 230. 正答率(テスト) = 0.2144
Generation: 240. 正答率(トレーニング) = 0.4
: 240. 正答率(テスト) = 0.2614
Generation: 250. 正答率(トレーニング) = 0.21
: 250. 正答率(テスト) = 0.2513
Generation: 260. 正答率(トレーニング) = 0.34
: 260. 正答率(テスト) = 0.2714
Generation: 270. 正答率(トレーニング) = 0.26
: 270. 正答率(テスト) = 0.2718
Generation: 280. 正答率(トレーニング) = 0.29
: 280. 正答率(テスト) = 0.2603
Generation: 290. 正答率(トレーニング) = 0.29
: 290. 正答率(テスト) = 0.2756
Generation: 300. 正答率(トレーニング) = 0.29
: 300. 正答率(テスト) = 0.2755
Generation: 310. 正答率(トレーニング) = 0.25
: 310. 正答率(テスト) = 0.2548
Generation: 320. 正答率(トレーニング) = 0.24
: 320. 正答率(テスト) = 0.2667
Generation: 330. 正答率(トレーニング) = 0.32
: 330. 正答率(テスト) = 0.2968
Generation: 340. 正答率(トレーニング) = 0.3
: 340. 正答率(テスト) = 0.2803
Generation: 350. 正答率(トレーニング) = 0.37
: 350. 正答率(テスト) = 0.2756
Generation: 360. 正答率(トレーニング) = 0.29
: 360. 正答率(テスト) = 0.2684
Generation: 370. 正答率(トレーニング) = 0.28
: 370. 正答率(テスト) = 0.2858
Generation: 380. 正答率(トレーニング) = 0.21
: 380. 正答率(テスト) = 0.2972
Generation: 390. 正答率(トレーニング) = 0.3
: 390. 正答率(テスト) = 0.2835
Generation: 400. 正答率(トレーニング) = 0.23
: 400. 正答率(テスト) = 0.3027
Generation: 410. 正答率(トレーニング) = 0.25
: 410. 正答率(テスト) = 0.2856

Generation: 420. 正答率(トレーニング) = 0.29
: 420. 正答率(テスト) = 0.2972
Generation: 430. 正答率(トレーニング) = 0.23
: 430. 正答率(テスト) = 0.2777
Generation: 440. 正答率(トレーニング) = 0.29
: 440. 正答率(テスト) = 0.306
Generation: 450. 正答率(トレーニング) = 0.28
: 450. 正答率(テスト) = 0.2945
Generation: 460. 正答率(トレーニング) = 0.26
: 460. 正答率(テスト) = 0.3176
Generation: 470. 正答率(トレーニング) = 0.35
: 470. 正答率(テスト) = 0.3194
Generation: 480. 正答率(トレーニング) = 0.3
: 480. 正答率(テスト) = 0.328
Generation: 490. 正答率(トレーニング) = 0.34
: 490. 正答率(テスト) = 0.3206
Generation: 500. 正答率(トレーニング) = 0.3
: 500. 正答率(テスト) = 0.33
Generation: 510. 正答率(トレーニング) = 0.39
: 510. 正答率(テスト) = 0.3217
Generation: 520. 正答率(トレーニング) = 0.36
: 520. 正答率(テスト) = 0.3455
Generation: 530. 正答率(トレーニング) = 0.38
: 530. 正答率(テスト) = 0.3528
Generation: 540. 正答率(トレーニング) = 0.43
: 540. 正答率(テスト) = 0.3407
Generation: 550. 正答率(トレーニング) = 0.41
: 550. 正答率(テスト) = 0.3712
Generation: 560. 正答率(トレーニング) = 0.32
: 560. 正答率(テスト) = 0.354
Generation: 570. 正答率(トレーニング) = 0.37
: 570. 正答率(テスト) = 0.3785
Generation: 580. 正答率(トレーニング) = 0.37
: 580. 正答率(テスト) = 0.3434
Generation: 590. 正答率(トレーニング) = 0.35
: 590. 正答率(テスト) = 0.401
Generation: 600. 正答率(トレーニング) = 0.47
: 600. 正答率(テスト) = 0.3798
Generation: 610. 正答率(トレーニング) = 0.4
: 610. 正答率(テスト) = 0.3758
Generation: 620. 正答率(トレーニング) = 0.37
: 620. 正答率(テスト) = 0.3925

Generation: 630. 正答率(トレーニング) = 0.33
: 630. 正答率(テスト) = 0.376
Generation: 640. 正答率(トレーニング) = 0.43
: 640. 正答率(テスト) = 0.3905
Generation: 650. 正答率(トレーニング) = 0.35
: 650. 正答率(テスト) = 0.3488
Generation: 660. 正答率(トレーニング) = 0.44
: 660. 正答率(テスト) = 0.3979
Generation: 670. 正答率(トレーニング) = 0.44
: 670. 正答率(テスト) = 0.3711
Generation: 680. 正答率(トレーニング) = 0.43
: 680. 正答率(テスト) = 0.3786
Generation: 690. 正答率(トレーニング) = 0.54
: 690. 正答率(テスト) = 0.421
Generation: 700. 正答率(トレーニング) = 0.44
: 700. 正答率(テスト) = 0.4281
Generation: 710. 正答率(トレーニング) = 0.45
: 710. 正答率(テスト) = 0.436
Generation: 720. 正答率(トレーニング) = 0.4
: 720. 正答率(テスト) = 0.4751
Generation: 730. 正答率(トレーニング) = 0.44
: 730. 正答率(テスト) = 0.4699
Generation: 740. 正答率(トレーニング) = 0.46
: 740. 正答率(テスト) = 0.4632
Generation: 750. 正答率(トレーニング) = 0.56
: 750. 正答率(テスト) = 0.4847
Generation: 760. 正答率(トレーニング) = 0.52
: 760. 正答率(テスト) = 0.5043
Generation: 770. 正答率(トレーニング) = 0.59
: 770. 正答率(テスト) = 0.5251
Generation: 780. 正答率(トレーニング) = 0.63
: 780. 正答率(テスト) = 0.5493
Generation: 790. 正答率(トレーニング) = 0.61
: 790. 正答率(テスト) = 0.5531
Generation: 800. 正答率(トレーニング) = 0.59
: 800. 正答率(テスト) = 0.5202
Generation: 810. 正答率(トレーニング) = 0.6
: 810. 正答率(テスト) = 0.5305
Generation: 820. 正答率(トレーニング) = 0.71
: 820. 正答率(テスト) = 0.5794
Generation: 830. 正答率(トレーニング) = 0.63
: 830. 正答率(テスト) = 0.5627

Generation: 840. 正答率(トレーニング) = 0.66
: 840. 正答率(テスト) = 0.5702
Generation: 850. 正答率(トレーニング) = 0.67
: 850. 正答率(テスト) = 0.5901
Generation: 860. 正答率(トレーニング) = 0.5
: 860. 正答率(テスト) = 0.5979
Generation: 870. 正答率(トレーニング) = 0.59
: 870. 正答率(テスト) = 0.5945
Generation: 880. 正答率(トレーニング) = 0.64
: 880. 正答率(テスト) = 0.6148
Generation: 890. 正答率(トレーニング) = 0.61
: 890. 正答率(テスト) = 0.6154
Generation: 900. 正答率(トレーニング) = 0.64
: 900. 正答率(テスト) = 0.6156
Generation: 910. 正答率(トレーニング) = 0.67
: 910. 正答率(テスト) = 0.648
Generation: 920. 正答率(トレーニング) = 0.61
: 920. 正答率(テスト) = 0.658
Generation: 930. 正答率(トレーニング) = 0.76
: 930. 正答率(テスト) = 0.697
Generation: 940. 正答率(トレーニング) = 0.71
: 940. 正答率(テスト) = 0.6941
Generation: 950. 正答率(トレーニング) = 0.68
: 950. 正答率(テスト) = 0.6829
Generation: 960. 正答率(トレーニング) = 0.78
: 960. 正答率(テスト) = 0.7369
Generation: 970. 正答率(トレーニング) = 0.73
: 970. 正答率(テスト) = 0.7317
Generation: 980. 正答率(トレーニング) = 0.69
: 980. 正答率(テスト) = 0.7117
Generation: 990. 正答率(トレーニング) = 0.81
: 990. 正答率(テスト) = 0.7445
Generation: 1000. 正答率(トレーニング) = 0.83
: 1000. 正答率(テスト) = 0.7495



[try] 活性化関数と重みの初期化方法を変えてみよう

SGD

```
In [34]: import sys, os
sys.path.append(os.pardir) # 親ディレクトリのファイルをインポートするための設定
import numpy as np
from collections import OrderedDict
from common import layers
from data.mnist import load_mnist
import matplotlib.pyplot as plt
from multi_layer_net import MultiLayerNet

# データの読み込み
(x_train, d_train), (x_test, d_test) = load_mnist(normalize=True, one_hot_label=True)

print("データ読み込み完了")

# batch_normalizationの設定 =====
#use_batchnorm = True
use_batchnorm = False
# =====

network = MultiLayerNet(input_size=784, hidden_size_list=[40, 20], output_size=10, activation='relu', weight_init_std='he',
                        use_batchnorm=use_batchnorm)

iters_num = 1000
train_size = x_train.shape[0]
batch_size = 100
learning_rate = 0.01

train_loss_list = []
accuracies_train = []
accuracies_test = []

plot_interval=10

for i in range(iters_num):
    idx_choice = np.random.choice(train_size, batch_size)
    x_batch = x_train[idx_choice]
    d_batch = d_train[idx_choice]

    # 勾配
```

```
grad = network.gradient(x_batch, d_batch)

for key in ('W1', 'W2', 'W3', 'b1', 'b2', 'b3'):
    network.params[key] -= learning_rate * grad[key]

    loss = network.loss(x_batch, d_batch)
    train_loss_list.append(loss)

if (i + 1) % plot_interval == 0:
    accr_test = network.accuracy(x_test, d_test)
    accuracies_test.append(accr_test)
    accr_train = network.accuracy(x_batch, d_batch)
    accuracies_train.append(accr_train)

    print('Generation: ' + str(i+1) + '. 正答率(トレーニング) = ' + str(accr_train))
    print('                : ' + str(i+1) + '. 正答率(テスト) = ' + str(accr_test))

lists = range(0, iters_num, plot_interval)
plt.plot(lists, accuracies_train, label="training set")
plt.plot(lists, accuracies_test, label="test set")
plt.legend(loc="lower right")
plt.title("accuracy")
plt.xlabel("count")
plt.ylabel("accuracy")
plt.ylim(0, 1.0)
# グラフの表示
plt.show()
```

データ読み込み完了

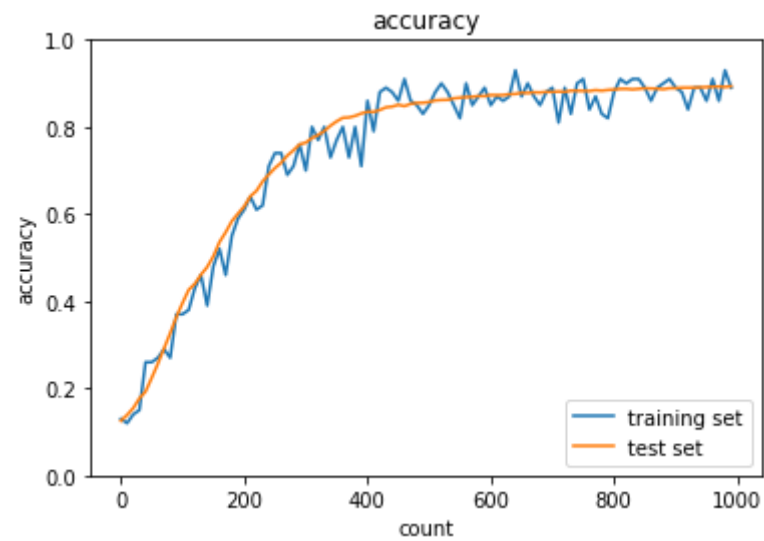
Generation: 10. 正答率(トレーニング) = 0.13
: 10. 正答率(テスト) = 0.1254
Generation: 20. 正答率(トレーニング) = 0.12
: 20. 正答率(テスト) = 0.1387
Generation: 30. 正答率(トレーニング) = 0.14
: 30. 正答率(テスト) = 0.1545
Generation: 40. 正答率(トレーニング) = 0.15
: 40. 正答率(テスト) = 0.1768
Generation: 50. 正答率(トレーニング) = 0.26
: 50. 正答率(テスト) = 0.1926
Generation: 60. 正答率(トレーニング) = 0.26
: 60. 正答率(テスト) = 0.2241
Generation: 70. 正答率(トレーニング) = 0.27
: 70. 正答率(テスト) = 0.2566
Generation: 80. 正答率(トレーニング) = 0.29
: 80. 正答率(テスト) = 0.2917
Generation: 90. 正答率(トレーニング) = 0.27
: 90. 正答率(テスト) = 0.3266
Generation: 100. 正答率(トレーニング) = 0.37
: 100. 正答率(テスト) = 0.3638
Generation: 110. 正答率(トレーニング) = 0.37
: 110. 正答率(テスト) = 0.3961
Generation: 120. 正答率(トレーニング) = 0.38
: 120. 正答率(テスト) = 0.4261
Generation: 130. 正答率(トレーニング) = 0.43
: 130. 正答率(テスト) = 0.4401
Generation: 140. 正答率(トレーニング) = 0.46
: 140. 正答率(テスト) = 0.461
Generation: 150. 正答率(トレーニング) = 0.39
: 150. 正答率(テスト) = 0.4785
Generation: 160. 正答率(トレーニング) = 0.48
: 160. 正答率(テスト) = 0.503
Generation: 170. 正答率(トレーニング) = 0.52
: 170. 正答率(テスト) = 0.5359
Generation: 180. 正答率(トレーニング) = 0.46
: 180. 正答率(テスト) = 0.5582
Generation: 190. 正答率(トレーニング) = 0.55
: 190. 正答率(テスト) = 0.5838
Generation: 200. 正答率(トレーニング) = 0.59
: 200. 正答率(テスト) = 0.601

Generation: 210. 正答率(トレーニング) = 0.61
: 210. 正答率(テスト) = 0.6176
Generation: 220. 正答率(トレーニング) = 0.64
: 220. 正答率(テスト) = 0.6407
Generation: 230. 正答率(トレーニング) = 0.61
: 230. 正答率(テスト) = 0.6534
Generation: 240. 正答率(トレーニング) = 0.62
: 240. 正答率(テスト) = 0.675
Generation: 250. 正答率(トレーニング) = 0.71
: 250. 正答率(テスト) = 0.6908
Generation: 260. 正答率(トレーニング) = 0.74
: 260. 正答率(テスト) = 0.7057
Generation: 270. 正答率(トレーニング) = 0.74
: 270. 正答率(テスト) = 0.7176
Generation: 280. 正答率(トレーニング) = 0.69
: 280. 正答率(テスト) = 0.7343
Generation: 290. 正答率(トレーニング) = 0.71
: 290. 正答率(テスト) = 0.7463
Generation: 300. 正答率(トレーニング) = 0.76
: 300. 正答率(テスト) = 0.7601
Generation: 310. 正答率(トレーニング) = 0.7
: 310. 正答率(テスト) = 0.764
Generation: 320. 正答率(トレーニング) = 0.8
: 320. 正答率(テスト) = 0.7746
Generation: 330. 正答率(トレーニング) = 0.77
: 330. 正答率(テスト) = 0.7815
Generation: 340. 正答率(トレーニング) = 0.8
: 340. 正答率(テスト) = 0.7905
Generation: 350. 正答率(トレーニング) = 0.73
: 350. 正答率(テスト) = 0.8029
Generation: 360. 正答率(トレーニング) = 0.77
: 360. 正答率(テスト) = 0.8128
Generation: 370. 正答率(トレーニング) = 0.8
: 370. 正答率(テスト) = 0.8209
Generation: 380. 正答率(トレーニング) = 0.73
: 380. 正答率(テスト) = 0.8218
Generation: 390. 正答率(トレーニング) = 0.8
: 390. 正答率(テスト) = 0.8254
Generation: 400. 正答率(トレーニング) = 0.71
: 400. 正答率(テスト) = 0.8309
Generation: 410. 正答率(トレーニング) = 0.86
: 410. 正答率(テスト) = 0.8349

Generation: 420. 正答率(トレーニング) = 0.79
: 420. 正答率(テスト) = 0.8345
Generation: 430. 正答率(トレーニング) = 0.88
: 430. 正答率(テスト) = 0.8389
Generation: 440. 正答率(トレーニング) = 0.89
: 440. 正答率(テスト) = 0.845
Generation: 450. 正答率(トレーニング) = 0.88
: 450. 正答率(テスト) = 0.8466
Generation: 460. 正答率(トレーニング) = 0.86
: 460. 正答率(テスト) = 0.8508
Generation: 470. 正答率(トレーニング) = 0.91
: 470. 正答率(テスト) = 0.8478
Generation: 480. 正答率(トレーニング) = 0.86
: 480. 正答率(テスト) = 0.853
Generation: 490. 正答率(トレーニング) = 0.85
: 490. 正答率(テスト) = 0.8551
Generation: 500. 正答率(トレーニング) = 0.83
: 500. 正答率(テスト) = 0.8554
Generation: 510. 正答率(トレーニング) = 0.85
: 510. 正答率(テスト) = 0.8572
Generation: 520. 正答率(トレーニング) = 0.88
: 520. 正答率(テスト) = 0.8609
Generation: 530. 正答率(トレーニング) = 0.9
: 530. 正答率(テスト) = 0.8619
Generation: 540. 正答率(トレーニング) = 0.88
: 540. 正答率(テスト) = 0.8622
Generation: 550. 正答率(トレーニング) = 0.85
: 550. 正答率(テスト) = 0.8656
Generation: 560. 正答率(トレーニング) = 0.82
: 560. 正答率(テスト) = 0.8671
Generation: 570. 正答率(トレーニング) = 0.9
: 570. 正答率(テスト) = 0.8685
Generation: 580. 正答率(トレーニング) = 0.85
: 580. 正答率(テスト) = 0.8691
Generation: 590. 正答率(トレーニング) = 0.87
: 590. 正答率(テスト) = 0.8708
Generation: 600. 正答率(トレーニング) = 0.89
: 600. 正答率(テスト) = 0.8708
Generation: 610. 正答率(トレーニング) = 0.85
: 610. 正答率(テスト) = 0.8732
Generation: 620. 正答率(トレーニング) = 0.87
: 620. 正答率(テスト) = 0.8729

Generation: 630. 正答率(トレーニング) = 0.86
: 630. 正答率(テスト) = 0.8731
Generation: 640. 正答率(トレーニング) = 0.87
: 640. 正答率(テスト) = 0.875
Generation: 650. 正答率(トレーニング) = 0.93
: 650. 正答率(テスト) = 0.8757
Generation: 660. 正答率(トレーニング) = 0.87
: 660. 正答率(テスト) = 0.8785
Generation: 670. 正答率(トレーニング) = 0.9
: 670. 正答率(テスト) = 0.8778
Generation: 680. 正答率(トレーニング) = 0.87
: 680. 正答率(テスト) = 0.8795
Generation: 690. 正答率(トレーニング) = 0.85
: 690. 正答率(テスト) = 0.879
Generation: 700. 正答率(トレーニング) = 0.88
: 700. 正答率(テスト) = 0.8805
Generation: 710. 正答率(トレーニング) = 0.89
: 710. 正答率(テスト) = 0.8803
Generation: 720. 正答率(トレーニング) = 0.81
: 720. 正答率(テスト) = 0.8813
Generation: 730. 正答率(トレーニング) = 0.89
: 730. 正答率(テスト) = 0.8804
Generation: 740. 正答率(トレーニング) = 0.83
: 740. 正答率(テスト) = 0.8832
Generation: 750. 正答率(トレーニング) = 0.9
: 750. 正答率(テスト) = 0.8829
Generation: 760. 正答率(トレーニング) = 0.91
: 760. 正答率(テスト) = 0.8828
Generation: 770. 正答率(トレーニング) = 0.84
: 770. 正答率(テスト) = 0.8824
Generation: 780. 正答率(トレーニング) = 0.87
: 780. 正答率(テスト) = 0.8845
Generation: 790. 正答率(トレーニング) = 0.83
: 790. 正答率(テスト) = 0.8832
Generation: 800. 正答率(トレーニング) = 0.82
: 800. 正答率(テスト) = 0.8849
Generation: 810. 正答率(トレーニング) = 0.88
: 810. 正答率(テスト) = 0.8859
Generation: 820. 正答率(トレーニング) = 0.91
: 820. 正答率(テスト) = 0.8872
Generation: 830. 正答率(トレーニング) = 0.9
: 830. 正答率(テスト) = 0.8874

Generation: 840. 正答率(トレーニング) = 0.91
: 840. 正答率(テスト) = 0.8862
Generation: 850. 正答率(トレーニング) = 0.91
: 850. 正答率(テスト) = 0.8874
Generation: 860. 正答率(トレーニング) = 0.89
: 860. 正答率(テスト) = 0.8888
Generation: 870. 正答率(トレーニング) = 0.86
: 870. 正答率(テスト) = 0.8882
Generation: 880. 正答率(トレーニング) = 0.89
: 880. 正答率(テスト) = 0.8879
Generation: 890. 正答率(トレーニング) = 0.9
: 890. 正答率(テスト) = 0.8866
Generation: 900. 正答率(トレーニング) = 0.91
: 900. 正答率(テスト) = 0.8889
Generation: 910. 正答率(トレーニング) = 0.89
: 910. 正答率(テスト) = 0.889
Generation: 920. 正答率(トレーニング) = 0.88
: 920. 正答率(テスト) = 0.89
Generation: 930. 正答率(トレーニング) = 0.84
: 930. 正答率(テスト) = 0.8901
Generation: 940. 正答率(トレーニング) = 0.89
: 940. 正答率(テスト) = 0.8902
Generation: 950. 正答率(トレーニング) = 0.89
: 950. 正答率(テスト) = 0.891
Generation: 960. 正答率(トレーニング) = 0.86
: 960. 正答率(テスト) = 0.8915
Generation: 970. 正答率(トレーニング) = 0.91
: 970. 正答率(テスト) = 0.8923
Generation: 980. 正答率(トレーニング) = 0.86
: 980. 正答率(テスト) = 0.8927
Generation: 990. 正答率(トレーニング) = 0.93
: 990. 正答率(テスト) = 0.8917
Generation: 1000. 正答率(トレーニング) = 0.89
: 1000. 正答率(テスト) = 0.8934



[try] バッチ正規化をしてみよう & ReLU, He

SGD

```
In [47]: import sys, os
sys.path.append(os.pardir) # 親ディレクトリのファイルをインポートするための設定
import numpy as np
from collections import OrderedDict
from common import layers
from data.mnist import load_mnist
import matplotlib.pyplot as plt
from multi_layer_net import MultiLayerNet

# データの読み込み
(x_train, d_train), (x_test, d_test) = load_mnist(normalize=True, one_hot_label=True)

print("データ読み込み完了")

# batch_normalizationの設定 =====
use_batchnorm = True
#use_batchnorm = False
# =====

network = MultiLayerNet(input_size=784, hidden_size_list=[40, 20], output_size=10, activation='relu', weight_init_std='he',
                        use_batchnorm=use_batchnorm)

iters_num = 1000
train_size = x_train.shape[0]
batch_size = 100
learning_rate = 0.01

train_loss_list = []
accuracies_train = []
accuracies_test = []

plot_interval=10

for i in range(iters_num):
    idx_choice = np.random.choice(train_size, batch_size)
    x_batch = x_train[idx_choice]
    d_batch = d_train[idx_choice]

    # 勾配
```

```
grad = network.gradient(x_batch, d_batch)

for key in ('W1', 'W2', 'W3', 'b1', 'b2', 'b3'):
    network.params[key] -= learning_rate * grad[key]

    loss = network.loss(x_batch, d_batch)
    train_loss_list.append(loss)

if (i + 1) % plot_interval == 0:
    accr_test = network.accuracy(x_test, d_test)
    accuracies_test.append(accr_test)
    accr_train = network.accuracy(x_batch, d_batch)
    accuracies_train.append(accr_train)

    print('Generation: ' + str(i+1) + '. 正答率(トレーニング) = ' + str(accr_train))
    print('                : ' + str(i+1) + '. 正答率(テスト) = ' + str(accr_test))

lists = range(0, iters_num, plot_interval)
plt.plot(lists, accuracies_train, label="training set")
plt.plot(lists, accuracies_test, label="test set")
plt.legend(loc="lower right")
plt.title("accuracy")
plt.xlabel("count")
plt.ylabel("accuracy")
plt.ylim(0, 1.0)
# グラフの表示
plt.show()
```

データ読み込み完了

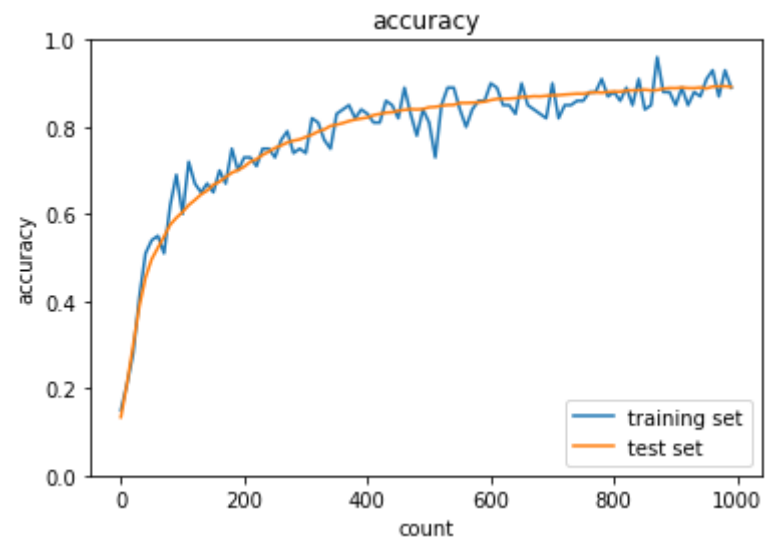
Generation: 10. 正答率(トレーニング) = 0.15
: 10. 正答率(テスト) = 0.1338
Generation: 20. 正答率(トレーニング) = 0.21
: 20. 正答率(テスト) = 0.2112
Generation: 30. 正答率(トレーニング) = 0.28
: 30. 正答率(テスト) = 0.2992
Generation: 40. 正答率(トレーニング) = 0.41
: 40. 正答率(テスト) = 0.3886
Generation: 50. 正答率(トレーニング) = 0.51
: 50. 正答率(テスト) = 0.4546
Generation: 60. 正答率(トレーニング) = 0.54
: 60. 正答率(テスト) = 0.4966
Generation: 70. 正答率(トレーニング) = 0.55
: 70. 正答率(テスト) = 0.5226
Generation: 80. 正答率(トレーニング) = 0.51
: 80. 正答率(テスト) = 0.5483
Generation: 90. 正答率(トレーニング) = 0.62
: 90. 正答率(テスト) = 0.5757
Generation: 100. 正答率(トレーニング) = 0.69
: 100. 正答率(テスト) = 0.5915
Generation: 110. 正答率(トレーニング) = 0.6
: 110. 正答率(テスト) = 0.6048
Generation: 120. 正答率(トレーニング) = 0.72
: 120. 正答率(テスト) = 0.6206
Generation: 130. 正答率(トレーニング) = 0.67
: 130. 正答率(テスト) = 0.6323
Generation: 140. 正答率(トレーニング) = 0.65
: 140. 正答率(テスト) = 0.645
Generation: 150. 正答率(トレーニング) = 0.67
: 150. 正答率(テスト) = 0.654
Generation: 160. 正答率(トレーニング) = 0.65
: 160. 正答率(テスト) = 0.6673
Generation: 170. 正答率(トレーニング) = 0.7
: 170. 正答率(テスト) = 0.6743
Generation: 180. 正答率(トレーニング) = 0.67
: 180. 正答率(テスト) = 0.6848
Generation: 190. 正答率(トレーニング) = 0.75
: 190. 正答率(テスト) = 0.6949
Generation: 200. 正答率(トレーニング) = 0.7
: 200. 正答率(テスト) = 0.7009

Generation: 210. 正答率(トレーニング) = 0.73
: 210. 正答率(テスト) = 0.7087
Generation: 220. 正答率(トレーニング) = 0.73
: 220. 正答率(テスト) = 0.7201
Generation: 230. 正答率(トレーニング) = 0.71
: 230. 正答率(テスト) = 0.726
Generation: 240. 正答率(トレーニング) = 0.75
: 240. 正答率(テスト) = 0.7359
Generation: 250. 正答率(トレーニング) = 0.75
: 250. 正答率(テスト) = 0.7433
Generation: 260. 正答率(トレーニング) = 0.73
: 260. 正答率(テスト) = 0.7518
Generation: 270. 正答率(トレーニング) = 0.77
: 270. 正答率(テスト) = 0.7575
Generation: 280. 正答率(トレーニング) = 0.79
: 280. 正答率(テスト) = 0.7643
Generation: 290. 正答率(トレーニング) = 0.74
: 290. 正答率(テスト) = 0.7686
Generation: 300. 正答率(トレーニング) = 0.75
: 300. 正答率(テスト) = 0.7713
Generation: 310. 正答率(トレーニング) = 0.74
: 310. 正答率(テスト) = 0.7769
Generation: 320. 正答率(トレーニング) = 0.82
: 320. 正答率(テスト) = 0.7823
Generation: 330. 正答率(トレーニング) = 0.81
: 330. 正答率(テスト) = 0.7895
Generation: 340. 正答率(トレーニング) = 0.77
: 340. 正答率(テスト) = 0.7943
Generation: 350. 正答率(トレーニング) = 0.75
: 350. 正答率(テスト) = 0.8023
Generation: 360. 正答率(トレーニング) = 0.83
: 360. 正答率(テスト) = 0.8053
Generation: 370. 正答率(トレーニング) = 0.84
: 370. 正答率(テスト) = 0.8088
Generation: 380. 正答率(トレーニング) = 0.85
: 380. 正答率(テスト) = 0.8134
Generation: 390. 正答率(トレーニング) = 0.82
: 390. 正答率(テスト) = 0.8169
Generation: 400. 正答率(トレーニング) = 0.84
: 400. 正答率(テスト) = 0.8193
Generation: 410. 正答率(トレーニング) = 0.83
: 410. 正答率(テスト) = 0.8217

Generation: 420. 正答率(トレーニング) = 0.81
: 420. 正答率(テスト) = 0.8271
Generation: 430. 正答率(トレーニング) = 0.81
: 430. 正答率(テスト) = 0.8299
Generation: 440. 正答率(トレーニング) = 0.86
: 440. 正答率(テスト) = 0.8328
Generation: 450. 正答率(トレーニング) = 0.85
: 450. 正答率(テスト) = 0.8337
Generation: 460. 正答率(トレーニング) = 0.82
: 460. 正答率(テスト) = 0.8373
Generation: 470. 正答率(トレーニング) = 0.89
: 470. 正答率(テスト) = 0.8391
Generation: 480. 正答率(トレーニング) = 0.83
: 480. 正答率(テスト) = 0.8409
Generation: 490. 正答率(トレーニング) = 0.78
: 490. 正答率(テスト) = 0.8401
Generation: 500. 正答率(トレーニング) = 0.84
: 500. 正答率(テスト) = 0.841
Generation: 510. 正答率(トレーニング) = 0.81
: 510. 正答率(テスト) = 0.8452
Generation: 520. 正答率(トレーニング) = 0.73
: 520. 正答率(テスト) = 0.846
Generation: 530. 正答率(トレーニング) = 0.85
: 530. 正答率(テスト) = 0.8485
Generation: 540. 正答率(トレーニング) = 0.89
: 540. 正答率(テスト) = 0.8503
Generation: 550. 正答率(トレーニング) = 0.89
: 550. 正答率(テスト) = 0.8506
Generation: 560. 正答率(トレーニング) = 0.84
: 560. 正答率(テスト) = 0.8551
Generation: 570. 正答率(トレーニング) = 0.8
: 570. 正答率(テスト) = 0.8551
Generation: 580. 正答率(トレーニング) = 0.84
: 580. 正答率(テスト) = 0.8556
Generation: 590. 正答率(トレーニング) = 0.86
: 590. 正答率(テスト) = 0.8563
Generation: 600. 正答率(トレーニング) = 0.86
: 600. 正答率(テスト) = 0.858
Generation: 610. 正答率(トレーニング) = 0.9
: 610. 正答率(テスト) = 0.8616
Generation: 620. 正答率(トレーニング) = 0.89
: 620. 正答率(テスト) = 0.8643

Generation: 630. 正答率(トレーニング) = 0.85
: 630. 正答率(テスト) = 0.8644
Generation: 640. 正答率(トレーニング) = 0.85
: 640. 正答率(テスト) = 0.8648
Generation: 650. 正答率(トレーニング) = 0.83
: 650. 正答率(テスト) = 0.8661
Generation: 660. 正答率(トレーニング) = 0.9
: 660. 正答率(テスト) = 0.8688
Generation: 670. 正答率(トレーニング) = 0.85
: 670. 正答率(テスト) = 0.8689
Generation: 680. 正答率(トレーニング) = 0.84
: 680. 正答率(テスト) = 0.8705
Generation: 690. 正答率(トレーニング) = 0.83
: 690. 正答率(テスト) = 0.87
Generation: 700. 正答率(トレーニング) = 0.82
: 700. 正答率(テスト) = 0.8714
Generation: 710. 正答率(トレーニング) = 0.9
: 710. 正答率(テスト) = 0.8726
Generation: 720. 正答率(トレーニング) = 0.82
: 720. 正答率(テスト) = 0.8729
Generation: 730. 正答率(トレーニング) = 0.85
: 730. 正答率(テスト) = 0.8742
Generation: 740. 正答率(トレーニング) = 0.85
: 740. 正答率(テスト) = 0.8753
Generation: 750. 正答率(トレーニング) = 0.86
: 750. 正答率(テスト) = 0.8763
Generation: 760. 正答率(トレーニング) = 0.86
: 760. 正答率(テスト) = 0.8761
Generation: 770. 正答率(トレーニング) = 0.88
: 770. 正答率(テスト) = 0.8784
Generation: 780. 正答率(トレーニング) = 0.88
: 780. 正答率(テスト) = 0.8795
Generation: 790. 正答率(トレーニング) = 0.91
: 790. 正答率(テスト) = 0.8786
Generation: 800. 正答率(トレーニング) = 0.87
: 800. 正答率(テスト) = 0.8804
Generation: 810. 正答率(トレーニング) = 0.88
: 810. 正答率(テスト) = 0.8811
Generation: 820. 正答率(トレーニング) = 0.86
: 820. 正答率(テスト) = 0.8809
Generation: 830. 正答率(トレーニング) = 0.89
: 830. 正答率(テスト) = 0.8834

Generation: 840. 正答率(トレーニング) = 0.85
: 840. 正答率(テスト) = 0.884
Generation: 850. 正答率(トレーニング) = 0.91
: 850. 正答率(テスト) = 0.8842
Generation: 860. 正答率(トレーニング) = 0.84
: 860. 正答率(テスト) = 0.8855
Generation: 870. 正答率(トレーニング) = 0.85
: 870. 正答率(テスト) = 0.8839
Generation: 880. 正答率(トレーニング) = 0.96
: 880. 正答率(テスト) = 0.8847
Generation: 890. 正答率(トレーニング) = 0.88
: 890. 正答率(テスト) = 0.8877
Generation: 900. 正答率(トレーニング) = 0.88
: 900. 正答率(テスト) = 0.8893
Generation: 910. 正答率(トレーニング) = 0.85
: 910. 正答率(テスト) = 0.889
Generation: 920. 正答率(トレーニング) = 0.89
: 920. 正答率(テスト) = 0.8906
Generation: 930. 正答率(トレーニング) = 0.85
: 930. 正答率(テスト) = 0.8892
Generation: 940. 正答率(トレーニング) = 0.88
: 940. 正答率(テスト) = 0.8891
Generation: 950. 正答率(トレーニング) = 0.87
: 950. 正答率(テスト) = 0.8905
Generation: 960. 正答率(トレーニング) = 0.91
: 960. 正答率(テスト) = 0.8884
Generation: 970. 正答率(トレーニング) = 0.93
: 970. 正答率(テスト) = 0.8926
Generation: 980. 正答率(トレーニング) = 0.87
: 980. 正答率(テスト) = 0.8932
Generation: 990. 正答率(トレーニング) = 0.93
: 990. 正答率(テスト) = 0.8935
Generation: 1000. 正答率(トレーニング) = 0.89
: 1000. 正答率(テスト) = 0.8919



Momentum

```

In [44]: # データの読み込み
(x_train, d_train), (x_test, d_test) = load_mnist(normalize=True, one_hot_label=True)

print("データ読み込み完了")

# batch_normalizationの設定 =====
#use_batchnorm = True
use_batchnorm = False
# =====

network = MultiLayerNet(input_size=784, hidden_size_list=[40, 20], output_size=10, activation='sigmoid', weight_init_std=0.01,
                        use_batchnorm=use_batchnorm)

iters_num = 1000
train_size = x_train.shape[0]
batch_size = 100
learning_rate = 0.01
# 慣性
momentum = 0.9

train_loss_list = []
accuracies_train = []
accuracies_test = []

plot_interval=10

for i in range(iters_num):
    idx_choice = np.random.choice(train_size, batch_size)
    x_batch = x_train[idx_choice]
    d_batch = d_train[idx_choice]

    # 勾配
    grad = network.gradient(x_batch, d_batch)
    if i == 0:
        v = {}
    for key in ('W1', 'W2', 'W3', 'b1', 'b2', 'b3'):
        if i == 0:
            v[key] = np.zeros_like(network.params[key])
            print("i=0", key, "v=", v[key])
        v[key] = momentum * v[key] - learning_rate * grad[key]
        network.params[key] += v[key]

```

```
    loss = network.loss(x_batch, d_batch)
    train_loss_list.append(loss)

    if (i + 1) % plot_interval == 0:
        accr_test = network.accuracy(x_test, d_test)
        accuracies_test.append(accr_test)
        accr_train = network.accuracy(x_batch, d_batch)
        accuracies_train.append(accr_train)

        print('Generation: ' + str(i+1) + '. 正答率(トレーニング) = ' + str(accr_train))
        print('                : ' + str(i+1) + '. 正答率(テスト) = ' + str(accr_test))

lists = range(0, iters_num, plot_interval)
plt.plot(lists, accuracies_train, label="training set")
plt.plot(lists, accuracies_test, label="test set")
plt.legend(loc="lower right")
plt.title("accuracy")
plt.xlabel("count")
plt.ylabel("accuracy")
plt.ylim(0, 1.0)
# グラフの表示
plt.show()
```

[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

37/169

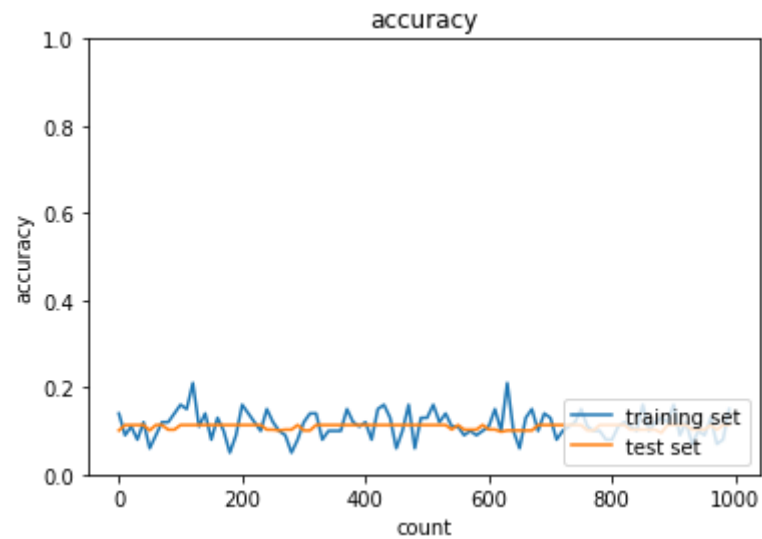
: 60. 正答率(テスト) = 0.101
Generation: 70. 正答率(トレーニング) = 0.09
 : 70. 正答率(テスト) = 0.1135
Generation: 80. 正答率(トレーニング) = 0.12
 : 80. 正答率(テスト) = 0.1135
Generation: 90. 正答率(トレーニング) = 0.12
 : 90. 正答率(テスト) = 0.1028
Generation: 100. 正答率(トレーニング) = 0.14
 : 100. 正答率(テスト) = 0.1028
Generation: 110. 正答率(トレーニング) = 0.16
 : 110. 正答率(テスト) = 0.1135
Generation: 120. 正答率(トレーニング) = 0.15
 : 120. 正答率(テスト) = 0.1135
Generation: 130. 正答率(トレーニング) = 0.21
 : 130. 正答率(テスト) = 0.1135
Generation: 140. 正答率(トレーニング) = 0.11
 : 140. 正答率(テスト) = 0.1135
Generation: 150. 正答率(トレーニング) = 0.14
 : 150. 正答率(テスト) = 0.1135
Generation: 160. 正答率(トレーニング) = 0.08
 : 160. 正答率(テスト) = 0.1135
Generation: 170. 正答率(トレーニング) = 0.13
 : 170. 正答率(テスト) = 0.1135
Generation: 180. 正答率(トレーニング) = 0.1
 : 180. 正答率(テスト) = 0.1135
Generation: 190. 正答率(トレーニング) = 0.05
 : 190. 正答率(テスト) = 0.1135
Generation: 200. 正答率(トレーニング) = 0.09
 : 200. 正答率(テスト) = 0.1135
Generation: 210. 正答率(トレーニング) = 0.16
 : 210. 正答率(テスト) = 0.1135
Generation: 220. 正答率(トレーニング) = 0.14
 : 220. 正答率(テスト) = 0.1135
Generation: 230. 正答率(トレーニング) = 0.12
 : 230. 正答率(テスト) = 0.1135
Generation: 240. 正答率(トレーニング) = 0.1
 : 240. 正答率(テスト) = 0.1135
Generation: 250. 正答率(トレーニング) = 0.15
 : 250. 正答率(テスト) = 0.1028
Generation: 260. 正答率(トレーニング) = 0.12
 : 260. 正答率(テスト) = 0.1028
Generation: 270. 正答率(トレーニング) = 0.1

: 270. 正答率(テスト) = 0.101
Generation: 280. 正答率(トレーニング) = 0.09
 : 280. 正答率(テスト) = 0.1028
Generation: 290. 正答率(トレーニング) = 0.05
 : 290. 正答率(テスト) = 0.1028
Generation: 300. 正答率(トレーニング) = 0.08
 : 300. 正答率(テスト) = 0.1135
Generation: 310. 正答率(トレーニング) = 0.12
 : 310. 正答率(テスト) = 0.1009
Generation: 320. 正答率(トレーニング) = 0.14
 : 320. 正答率(テスト) = 0.1009
Generation: 330. 正答率(トレーニング) = 0.14
 : 330. 正答率(テスト) = 0.1135
Generation: 340. 正答率(トレーニング) = 0.08
 : 340. 正答率(テスト) = 0.1135
Generation: 350. 正答率(トレーニング) = 0.1
 : 350. 正答率(テスト) = 0.1135
Generation: 360. 正答率(トレーニング) = 0.1
 : 360. 正答率(テスト) = 0.1135
Generation: 370. 正答率(トレーニング) = 0.1
 : 370. 正答率(テスト) = 0.1135
Generation: 380. 正答率(トレーニング) = 0.15
 : 380. 正答率(テスト) = 0.1135
Generation: 390. 正答率(トレーニング) = 0.12
 : 390. 正答率(テスト) = 0.1135
Generation: 400. 正答率(トレーニング) = 0.11
 : 400. 正答率(テスト) = 0.1135
Generation: 410. 正答率(トレーニング) = 0.12
 : 410. 正答率(テスト) = 0.1135
Generation: 420. 正答率(トレーニング) = 0.08
 : 420. 正答率(テスト) = 0.1135
Generation: 430. 正答率(トレーニング) = 0.15
 : 430. 正答率(テスト) = 0.1135
Generation: 440. 正答率(トレーニング) = 0.16
 : 440. 正答率(テスト) = 0.1135
Generation: 450. 正答率(トレーニング) = 0.13
 : 450. 正答率(テスト) = 0.1135
Generation: 460. 正答率(トレーニング) = 0.06
 : 460. 正答率(テスト) = 0.1135
Generation: 470. 正答率(トレーニング) = 0.1
 : 470. 正答率(テスト) = 0.1135
Generation: 480. 正答率(トレーニング) = 0.16

: 480. 正答率(テスト) = 0.1135
Generation: 490. 正答率(トレーニング) = 0.06
 : 490. 正答率(テスト) = 0.1135
Generation: 500. 正答率(トレーニング) = 0.13
 : 500. 正答率(テスト) = 0.1135
Generation: 510. 正答率(トレーニング) = 0.13
 : 510. 正答率(テスト) = 0.1135
Generation: 520. 正答率(トレーニング) = 0.16
 : 520. 正答率(テスト) = 0.1135
Generation: 530. 正答率(トレーニング) = 0.12
 : 530. 正答率(テスト) = 0.1135
Generation: 540. 正答率(トレーニング) = 0.14
 : 540. 正答率(テスト) = 0.1135
Generation: 550. 正答率(トレーニング) = 0.11
 : 550. 正答率(テスト) = 0.1032
Generation: 560. 正答率(トレーニング) = 0.11
 : 560. 正答率(テスト) = 0.1135
Generation: 570. 正答率(トレーニング) = 0.09
 : 570. 正答率(テスト) = 0.1028
Generation: 580. 正答率(トレーニング) = 0.1
 : 580. 正答率(テスト) = 0.1028
Generation: 590. 正答率(トレーニング) = 0.09
 : 590. 正答率(テスト) = 0.1028
Generation: 600. 正答率(トレーニング) = 0.1
 : 600. 正答率(テスト) = 0.1135
Generation: 610. 正答率(トレーニング) = 0.11
 : 610. 正答率(テスト) = 0.1032
Generation: 620. 正答率(トレーニング) = 0.15
 : 620. 正答率(テスト) = 0.1032
Generation: 630. 正答率(トレーニング) = 0.1
 : 630. 正答率(テスト) = 0.098
Generation: 640. 正答率(トレーニング) = 0.21
 : 640. 正答率(テスト) = 0.101
Generation: 650. 正答率(トレーニング) = 0.1
 : 650. 正答率(テスト) = 0.101
Generation: 660. 正答率(トレーニング) = 0.06
 : 660. 正答率(テスト) = 0.101
Generation: 670. 正答率(トレーニング) = 0.13
 : 670. 正答率(テスト) = 0.1009
Generation: 680. 正答率(トレーニング) = 0.15
 : 680. 正答率(テスト) = 0.1009
Generation: 690. 正答率(トレーニング) = 0.1

: 690. 正答率(テスト) = 0.1135
Generation: 700. 正答率(トレーニング) = 0.14
 : 700. 正答率(テスト) = 0.1135
Generation: 710. 正答率(トレーニング) = 0.13
 : 710. 正答率(テスト) = 0.1135
Generation: 720. 正答率(トレーニング) = 0.08
 : 720. 正答率(テスト) = 0.1135
Generation: 730. 正答率(トレーニング) = 0.1
 : 730. 正答率(テスト) = 0.1135
Generation: 740. 正答率(トレーニング) = 0.11
 : 740. 正答率(テスト) = 0.1135
Generation: 750. 正答率(トレーニング) = 0.12
 : 750. 正答率(テスト) = 0.1135
Generation: 760. 正答率(トレーニング) = 0.15
 : 760. 正答率(テスト) = 0.1135
Generation: 770. 正答率(トレーニング) = 0.11
 : 770. 正答率(テスト) = 0.101
Generation: 780. 正答率(トレーニング) = 0.1
 : 780. 正答率(テスト) = 0.101
Generation: 790. 正答率(トレーニング) = 0.1
 : 790. 正答率(テスト) = 0.1135
Generation: 800. 正答率(トレーニング) = 0.08
 : 800. 正答率(テスト) = 0.1135
Generation: 810. 正答率(トレーニング) = 0.08
 : 810. 正答率(テスト) = 0.1135
Generation: 820. 正答率(トレーニング) = 0.11
 : 820. 正答率(テスト) = 0.1135
Generation: 830. 正答率(トレーニング) = 0.12
 : 830. 正答率(テスト) = 0.1135
Generation: 840. 正答率(トレーニング) = 0.11
 : 840. 正答率(テスト) = 0.1028
Generation: 850. 正答率(トレーニング) = 0.1
 : 850. 正答率(テスト) = 0.1028
Generation: 860. 正答率(トレーニング) = 0.16
 : 860. 正答率(テスト) = 0.1028
Generation: 870. 正答率(トレーニング) = 0.1
 : 870. 正答率(テスト) = 0.1028
Generation: 880. 正答率(トレーニング) = 0.12
 : 880. 正答率(テスト) = 0.1028
Generation: 890. 正答率(トレーニング) = 0.13
 : 890. 正答率(テスト) = 0.0958
Generation: 900. 正答率(トレーニング) = 0.12

: 900. 正答率(テスト) = 0.1135
Generation: 910. 正答率(トレーニング) = 0.16
 : 910. 正答率(テスト) = 0.1135
Generation: 920. 正答率(トレーニング) = 0.09
 : 920. 正答率(テスト) = 0.1135
Generation: 930. 正答率(トレーニング) = 0.12
 : 930. 正答率(テスト) = 0.1135
Generation: 940. 正答率(トレーニング) = 0.07
 : 940. 正答率(テスト) = 0.1028
Generation: 950. 正答率(トレーニング) = 0.1
 : 950. 正答率(テスト) = 0.1028
Generation: 960. 正答率(トレーニング) = 0.09
 : 960. 正答率(テスト) = 0.1028
Generation: 970. 正答率(トレーニング) = 0.13
 : 970. 正答率(テスト) = 0.1135
Generation: 980. 正答率(トレーニング) = 0.07
 : 980. 正答率(テスト) = 0.1028
Generation: 990. 正答率(トレーニング) = 0.08
 : 990. 正答率(テスト) = 0.1135
Generation: 1000. 正答率(トレーニング) = 0.15
 : 1000. 正答率(テスト) = 0.1135



[try] 学習率を変えてみよう

Momentum

```

In [46]: # データの読み込み
(x_train, d_train), (x_test, d_test) = load_mnist(normalize=True, one_hot_label=True)

print("データ読み込み完了")

# batch_normalizationの設定 =====
#use_batchnorm = True
use_batchnorm = False
# =====

network = MultiLayerNet(input_size=784, hidden_size_list=[40, 20], output_size=10, activation='sigmoid', weight_init_std=0.01,
                        use_batchnorm=use_batchnorm)

iters_num = 1000
train_size = x_train.shape[0]
batch_size = 100
learning_rate = 1
# 慣性
momentum = 0.9

train_loss_list = []
accuracies_train = []
accuracies_test = []

plot_interval=10

for i in range(iters_num):
    idx_choice = np.random.choice(train_size, batch_size)
    x_batch = x_train[idx_choice]
    d_batch = d_train[idx_choice]

    # 勾配
    grad = network.gradient(x_batch, d_batch)
    if i == 0:
        v = {}
    for key in ('W1', 'W2', 'W3', 'b1', 'b2', 'b3'):
        if i == 0:
            v[key] = np.zeros_like(network.params[key])
            print("i=0", key, "v=", v[key])
        v[key] = momentum * v[key] - learning_rate * grad[key]
        network.params[key] += v[key]

```

```
    loss = network.loss(x_batch, d_batch)
    train_loss_list.append(loss)

    if (i + 1) % plot_interval == 0:
        accr_test = network.accuracy(x_test, d_test)
        accuracies_test.append(accr_test)
        accr_train = network.accuracy(x_batch, d_batch)
        accuracies_train.append(accr_train)

        print('Generation: ' + str(i+1) + '. 正答率(トレーニング) = ' + str(accr_train))
        print('                : ' + str(i+1) + '. 正答率(テスト) = ' + str(accr_test))

lists = range(0, iters_num, plot_interval)
plt.plot(lists, accuracies_train, label="training set")
plt.plot(lists, accuracies_test, label="test set")
plt.legend(loc="lower right")
plt.title("accuracy")
plt.xlabel("count")
plt.ylabel("accuracy")
plt.ylim(0, 1.0)
# グラフの表示
plt.show()
```

[illegible]

file:///C:/Users/克扩/Desktop/upload/2_4_optimizer-ensyu.html

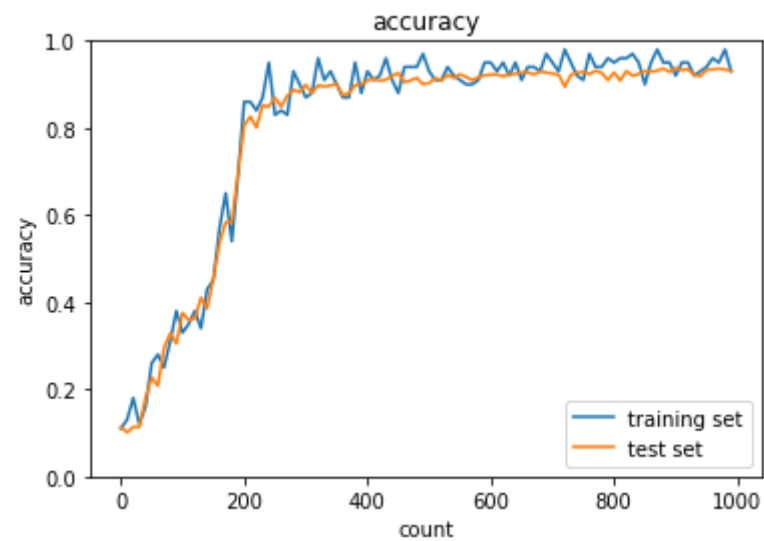
: 60. 正答率(テスト) = 0.2275
Generation: 70. 正答率(トレーニング) = 0.28
 : 70. 正答率(テスト) = 0.2078
Generation: 80. 正答率(トレーニング) = 0.25
 : 80. 正答率(テスト) = 0.2948
Generation: 90. 正答率(トレーニング) = 0.31
 : 90. 正答率(テスト) = 0.3283
Generation: 100. 正答率(トレーニング) = 0.38
 : 100. 正答率(テスト) = 0.3049
Generation: 110. 正答率(トレーニング) = 0.33
 : 110. 正答率(テスト) = 0.3754
Generation: 120. 正答率(トレーニング) = 0.35
 : 120. 正答率(テスト) = 0.3588
Generation: 130. 正答率(トレーニング) = 0.38
 : 130. 正答率(テスト) = 0.3626
Generation: 140. 正答率(トレーニング) = 0.34
 : 140. 正答率(テスト) = 0.4103
Generation: 150. 正答率(トレーニング) = 0.43
 : 150. 正答率(テスト) = 0.3858
Generation: 160. 正答率(トレーニング) = 0.45
 : 160. 正答率(テスト) = 0.4543
Generation: 170. 正答率(トレーニング) = 0.57
 : 170. 正答率(テスト) = 0.5364
Generation: 180. 正答率(トレーニング) = 0.65
 : 180. 正答率(テスト) = 0.5828
Generation: 190. 正答率(トレーニング) = 0.54
 : 190. 正答率(テスト) = 0.5837
Generation: 200. 正答率(トレーニング) = 0.69
 : 200. 正答率(テスト) = 0.6947
Generation: 210. 正答率(トレーニング) = 0.86
 : 210. 正答率(テスト) = 0.8056
Generation: 220. 正答率(トレーニング) = 0.86
 : 220. 正答率(テスト) = 0.8261
Generation: 230. 正答率(トレーニング) = 0.84
 : 230. 正答率(テスト) = 0.8009
Generation: 240. 正答率(トレーニング) = 0.87
 : 240. 正答率(テスト) = 0.8516
Generation: 250. 正答率(トレーニング) = 0.95
 : 250. 正答率(テスト) = 0.8488
Generation: 260. 正答率(トレーニング) = 0.83
 : 260. 正答率(テスト) = 0.8693
Generation: 270. 正答率(トレーニング) = 0.84

: 270. 正答率(テスト) = 0.8488
Generation: 280. 正答率(トレーニング) = 0.83
 : 280. 正答率(テスト) = 0.8738
Generation: 290. 正答率(トレーニング) = 0.93
 : 290. 正答率(テスト) = 0.8873
Generation: 300. 正答率(トレーニング) = 0.9
 : 300. 正答率(テスト) = 0.8819
Generation: 310. 正答率(トレーニング) = 0.87
 : 310. 正答率(テスト) = 0.8991
Generation: 320. 正答率(トレーニング) = 0.88
 : 320. 正答率(テスト) = 0.88
Generation: 330. 正答率(トレーニング) = 0.96
 : 330. 正答率(テスト) = 0.8978
Generation: 340. 正答率(トレーニング) = 0.91
 : 340. 正答率(テスト) = 0.8952
Generation: 350. 正答率(トレーニング) = 0.93
 : 350. 正答率(テスト) = 0.8972
Generation: 360. 正答率(トレーニング) = 0.9
 : 360. 正答率(テスト) = 0.9011
Generation: 370. 正答率(トレーニング) = 0.87
 : 370. 正答率(テスト) = 0.8749
Generation: 380. 正答率(トレーニング) = 0.87
 : 380. 正答率(テスト) = 0.8796
Generation: 390. 正答率(トレーニング) = 0.95
 : 390. 正答率(テスト) = 0.8982
Generation: 400. 正答率(トレーニング) = 0.88
 : 400. 正答率(テスト) = 0.899
Generation: 410. 正答率(トレーニング) = 0.93
 : 410. 正答率(テスト) = 0.9089
Generation: 420. 正答率(トレーニング) = 0.91
 : 420. 正答率(テスト) = 0.9122
Generation: 430. 正答率(トレーニング) = 0.92
 : 430. 正答率(テスト) = 0.9084
Generation: 440. 正答率(トレーニング) = 0.96
 : 440. 正答率(テスト) = 0.9107
Generation: 450. 正答率(トレーニング) = 0.91
 : 450. 正答率(テスト) = 0.9189
Generation: 460. 正答率(トレーニング) = 0.88
 : 460. 正答率(テスト) = 0.926
Generation: 470. 正答率(トレーニング) = 0.94
 : 470. 正答率(テスト) = 0.905
Generation: 480. 正答率(トレーニング) = 0.94

: 480. 正答率(テスト) = 0.9089
Generation: 490. 正答率(トレーニング) = 0.94
 : 490. 正答率(テスト) = 0.9147
Generation: 500. 正答率(トレーニング) = 0.97
 : 500. 正答率(テスト) = 0.9005
Generation: 510. 正答率(トレーニング) = 0.93
 : 510. 正答率(テスト) = 0.903
Generation: 520. 正答率(トレーニング) = 0.91
 : 520. 正答率(テスト) = 0.9142
Generation: 530. 正答率(トレーニング) = 0.91
 : 530. 正答率(テスト) = 0.9089
Generation: 540. 正答率(トレーニング) = 0.94
 : 540. 正答率(テスト) = 0.9215
Generation: 550. 正答率(トレーニング) = 0.92
 : 550. 正答率(テスト) = 0.9148
Generation: 560. 正答率(トレーニング) = 0.91
 : 560. 正答率(テスト) = 0.9229
Generation: 570. 正答率(トレーニング) = 0.9
 : 570. 正答率(テスト) = 0.917
Generation: 580. 正答率(トレーニング) = 0.9
 : 580. 正答率(テスト) = 0.9095
Generation: 590. 正答率(トレーニング) = 0.91
 : 590. 正答率(テスト) = 0.9163
Generation: 600. 正答率(トレーニング) = 0.95
 : 600. 正答率(テスト) = 0.921
Generation: 610. 正答率(トレーニング) = 0.95
 : 610. 正答率(テスト) = 0.9222
Generation: 620. 正答率(トレーニング) = 0.93
 : 620. 正答率(テスト) = 0.9226
Generation: 630. 正答率(トレーニング) = 0.95
 : 630. 正答率(テスト) = 0.9185
Generation: 640. 正答率(トレーニング) = 0.92
 : 640. 正答率(テスト) = 0.9225
Generation: 650. 正答率(トレーニング) = 0.95
 : 650. 正答率(テスト) = 0.9242
Generation: 660. 正答率(トレーニング) = 0.91
 : 660. 正答率(テスト) = 0.9267
Generation: 670. 正答率(トレーニング) = 0.94
 : 670. 正答率(テスト) = 0.9276
Generation: 680. 正答率(トレーニング) = 0.94
 : 680. 正答率(テスト) = 0.9225
Generation: 690. 正答率(トレーニング) = 0.93

: 690. 正答率(テスト) = 0.9291
Generation: 700. 正答率(トレーニング) = 0.97
 : 700. 正答率(テスト) = 0.9271
Generation: 710. 正答率(トレーニング) = 0.95
 : 710. 正答率(テスト) = 0.9248
Generation: 720. 正答率(トレーニング) = 0.93
 : 720. 正答率(テスト) = 0.9222
Generation: 730. 正答率(トレーニング) = 0.98
 : 730. 正答率(テスト) = 0.895
Generation: 740. 正答率(トレーニング) = 0.95
 : 740. 正答率(テスト) = 0.9217
Generation: 750. 正答率(トレーニング) = 0.92
 : 750. 正答率(テスト) = 0.9266
Generation: 760. 正答率(トレーニング) = 0.91
 : 760. 正答率(テスト) = 0.9288
Generation: 770. 正答率(トレーニング) = 0.97
 : 770. 正答率(テスト) = 0.9232
Generation: 780. 正答率(トレーニング) = 0.94
 : 780. 正答率(テスト) = 0.9311
Generation: 790. 正答率(トレーニング) = 0.94
 : 790. 正答率(テスト) = 0.9265
Generation: 800. 正答率(トレーニング) = 0.96
 : 800. 正答率(テスト) = 0.9097
Generation: 810. 正答率(トレーニング) = 0.95
 : 810. 正答率(テスト) = 0.9269
Generation: 820. 正答率(トレーニング) = 0.96
 : 820. 正答率(テスト) = 0.9088
Generation: 830. 正答率(トレーニング) = 0.96
 : 830. 正答率(テスト) = 0.9305
Generation: 840. 正答率(トレーニング) = 0.97
 : 840. 正答率(テスト) = 0.9196
Generation: 850. 正答率(トレーニング) = 0.95
 : 850. 正答率(テスト) = 0.9238
Generation: 860. 正答率(トレーニング) = 0.9
 : 860. 正答率(テスト) = 0.9308
Generation: 870. 正答率(トレーニング) = 0.95
 : 870. 正答率(テスト) = 0.9289
Generation: 880. 正答率(トレーニング) = 0.98
 : 880. 正答率(テスト) = 0.9313
Generation: 890. 正答率(トレーニング) = 0.95
 : 890. 正答率(テスト) = 0.9363
Generation: 900. 正答率(トレーニング) = 0.95

: 900. 正答率(テスト) = 0.9283
Generation: 910. 正答率(トレーニング) = 0.92
 : 910. 正答率(テスト) = 0.9377
Generation: 920. 正答率(トレーニング) = 0.95
 : 920. 正答率(テスト) = 0.933
Generation: 930. 正答率(トレーニング) = 0.95
 : 930. 正答率(テスト) = 0.9344
Generation: 940. 正答率(トレーニング) = 0.92
 : 940. 正答率(テスト) = 0.922
Generation: 950. 正答率(トレーニング) = 0.93
 : 950. 正答率(テスト) = 0.9183
Generation: 960. 正答率(トレーニング) = 0.94
 : 960. 正答率(テスト) = 0.9332
Generation: 970. 正答率(トレーニング) = 0.96
 : 970. 正答率(テスト) = 0.9344
Generation: 980. 正答率(トレーニング) = 0.95
 : 980. 正答率(テスト) = 0.9358
Generation: 990. 正答率(トレーニング) = 0.98
 : 990. 正答率(テスト) = 0.9345
Generation: 1000. 正答率(トレーニング) = 0.93
 : 1000. 正答率(テスト) = 0.9303



[try] 活性化関数と重みの初期化方法を変えてみよう

Momentum

```

In [36]: # データの読み込み
(x_train, d_train), (x_test, d_test) = load_mnist(normalize=True, one_hot_label=True)

print("データ読み込み完了")

# batch_normalizationの設定 =====
#use_batchnorm = True
use_batchnorm = False
# =====

network = MultiLayerNet(input_size=784, hidden_size_list=[40, 20], output_size=10, activation='relu', weight_init_std='he',
                        use_batchnorm=use_batchnorm)

iters_num = 1000
train_size = x_train.shape[0]
batch_size = 100
learning_rate = 0.01
# 慣性
momentum = 0.9

train_loss_list = []
accuracies_train = []
accuracies_test = []

plot_interval=10

for i in range(iters_num):
    idx_choice = np.random.choice(train_size, batch_size)
    x_batch = x_train[idx_choice]
    d_batch = d_train[idx_choice]

    # 勾配
    grad = network.gradient(x_batch, d_batch)
    if i == 0:
        v = {}
    for key in ('W1', 'W2', 'W3', 'b1', 'b2', 'b3'):
        if i == 0:
            v[key] = np.zeros_like(network.params[key])
            print("i=0", key, "v=", v[key])
        v[key] = momentum * v[key] - learning_rate * grad[key]
        network.params[key] += v[key]

```

```
    loss = network.loss(x_batch, d_batch)
    train_loss_list.append(loss)

    if (i + 1) % plot_interval == 0:
        accr_test = network.accuracy(x_test, d_test)
        accuracies_test.append(accr_test)
        accr_train = network.accuracy(x_batch, d_batch)
        accuracies_train.append(accr_train)

        print('Generation: ' + str(i+1) + '. 正答率(トレーニング) = ' + str(accr_train))
        print('                : ' + str(i+1) + '. 正答率(テスト) = ' + str(accr_test))

lists = range(0, iters_num, plot_interval)
plt.plot(lists, accuracies_train, label="training set")
plt.plot(lists, accuracies_test, label="test set")
plt.legend(loc="lower right")
plt.title("accuracy")
plt.xlabel("count")
plt.ylabel("accuracy")
plt.ylim(0, 1.0)
# グラフの表示
plt.show()
```

[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

file:///C:/Users/克扩/Desktop/upload/2_4_optimizer-ensyu.html

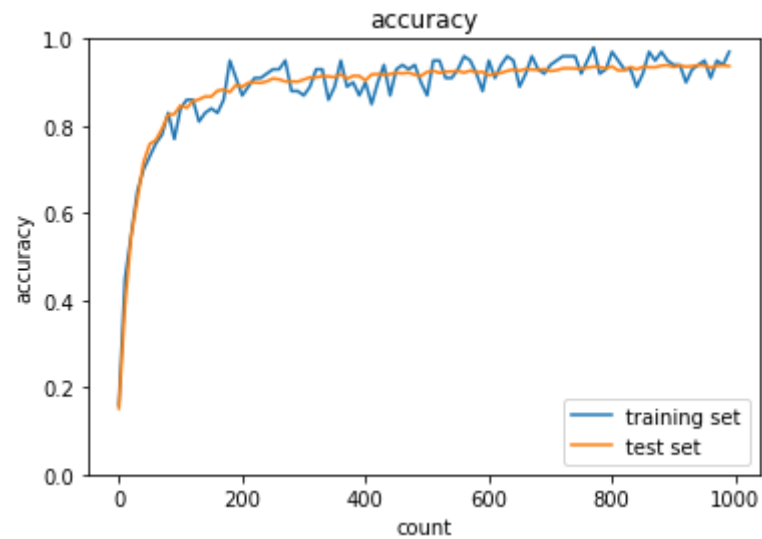
: 60. 正答率(テスト) = 0.7582
Generation: 70. 正答率(トレーニング) = 0.76
 : 70. 正答率(テスト) = 0.7675
Generation: 80. 正答率(トレーニング) = 0.78
 : 80. 正答率(テスト) = 0.7934
Generation: 90. 正答率(トレーニング) = 0.83
 : 90. 正答率(テスト) = 0.8251
Generation: 100. 正答率(トレーニング) = 0.77
 : 100. 正答率(テスト) = 0.8271
Generation: 110. 正答率(トレーニング) = 0.84
 : 110. 正答率(テスト) = 0.8471
Generation: 120. 正答率(トレーニング) = 0.86
 : 120. 正答率(テスト) = 0.8414
Generation: 130. 正答率(トレーニング) = 0.86
 : 130. 正答率(テスト) = 0.8565
Generation: 140. 正答率(トレーニング) = 0.81
 : 140. 正答率(テスト) = 0.861
Generation: 150. 正答率(トレーニング) = 0.83
 : 150. 正答率(テスト) = 0.8674
Generation: 160. 正答率(トレーニング) = 0.84
 : 160. 正答率(テスト) = 0.8673
Generation: 170. 正答率(トレーニング) = 0.83
 : 170. 正答率(テスト) = 0.8808
Generation: 180. 正答率(トレーニング) = 0.86
 : 180. 正答率(テスト) = 0.8835
Generation: 190. 正答率(トレーニング) = 0.95
 : 190. 正答率(テスト) = 0.8776
Generation: 200. 正答率(トレーニング) = 0.91
 : 200. 正答率(テスト) = 0.894
Generation: 210. 正答率(トレーニング) = 0.87
 : 210. 正答率(テスト) = 0.8899
Generation: 220. 正答率(トレーニング) = 0.89
 : 220. 正答率(テスト) = 0.8988
Generation: 230. 正答率(トレーニング) = 0.91
 : 230. 正答率(テスト) = 0.9002
Generation: 240. 正答率(トレーニング) = 0.91
 : 240. 正答率(テスト) = 0.8987
Generation: 250. 正答率(トレーニング) = 0.92
 : 250. 正答率(テスト) = 0.9023
Generation: 260. 正答率(トレーニング) = 0.93
 : 260. 正答率(テスト) = 0.9088
Generation: 270. 正答率(トレーニング) = 0.93

: 270. 正答率(テスト) = 0.9069
Generation: 280. 正答率(トレーニング) = 0.95
 : 280. 正答率(テスト) = 0.9019
Generation: 290. 正答率(トレーニング) = 0.88
 : 290. 正答率(テスト) = 0.9024
Generation: 300. 正答率(トレーニング) = 0.88
 : 300. 正答率(テスト) = 0.9016
Generation: 310. 正答率(トレーニング) = 0.87
 : 310. 正答率(テスト) = 0.9064
Generation: 320. 正答率(トレーニング) = 0.89
 : 320. 正答率(テスト) = 0.9114
Generation: 330. 正答率(トレーニング) = 0.93
 : 330. 正答率(テスト) = 0.9112
Generation: 340. 正答率(トレーニング) = 0.93
 : 340. 正答率(テスト) = 0.9145
Generation: 350. 正答率(トレーニング) = 0.86
 : 350. 正答率(テスト) = 0.9136
Generation: 360. 正答率(トレーニング) = 0.89
 : 360. 正答率(テスト) = 0.9118
Generation: 370. 正答率(トレーニング) = 0.95
 : 370. 正答率(テスト) = 0.9182
Generation: 380. 正答率(トレーニング) = 0.89
 : 380. 正答率(テスト) = 0.9083
Generation: 390. 正答率(トレーニング) = 0.9
 : 390. 正答率(テスト) = 0.9149
Generation: 400. 正答率(トレーニング) = 0.87
 : 400. 正答率(テスト) = 0.9147
Generation: 410. 正答率(トレーニング) = 0.9
 : 410. 正答率(テスト) = 0.9044
Generation: 420. 正答率(トレーニング) = 0.85
 : 420. 正答率(テスト) = 0.9183
Generation: 430. 正答率(トレーニング) = 0.9
 : 430. 正答率(テスト) = 0.9189
Generation: 440. 正答率(トレーニング) = 0.94
 : 440. 正答率(テスト) = 0.9168
Generation: 450. 正答率(トレーニング) = 0.87
 : 450. 正答率(テスト) = 0.9191
Generation: 460. 正答率(トレーニング) = 0.93
 : 460. 正答率(テスト) = 0.9218
Generation: 470. 正答率(トレーニング) = 0.94
 : 470. 正答率(テスト) = 0.9207
Generation: 480. 正答率(トレーニング) = 0.93

: 480. 正答率(テスト) = 0.9222
Generation: 490. 正答率(トレーニング) = 0.94
 : 490. 正答率(テスト) = 0.9174
Generation: 500. 正答率(トレーニング) = 0.9
 : 500. 正答率(テスト) = 0.9148
Generation: 510. 正答率(トレーニング) = 0.87
 : 510. 正答率(テスト) = 0.9239
Generation: 520. 正答率(トレーニング) = 0.95
 : 520. 正答率(テスト) = 0.9264
Generation: 530. 正答率(トレーニング) = 0.95
 : 530. 正答率(テスト) = 0.9218
Generation: 540. 正答率(トレーニング) = 0.91
 : 540. 正答率(テスト) = 0.9246
Generation: 550. 正答率(トレーニング) = 0.91
 : 550. 正答率(テスト) = 0.9251
Generation: 560. 正答率(トレーニング) = 0.93
 : 560. 正答率(テスト) = 0.9258
Generation: 570. 正答率(トレーニング) = 0.96
 : 570. 正答率(テスト) = 0.9224
Generation: 580. 正答率(トレーニング) = 0.95
 : 580. 正答率(テスト) = 0.9269
Generation: 590. 正答率(トレーニング) = 0.92
 : 590. 正答率(テスト) = 0.923
Generation: 600. 正答率(トレーニング) = 0.88
 : 600. 正答率(テスト) = 0.9242
Generation: 610. 正答率(トレーニング) = 0.95
 : 610. 正答率(テスト) = 0.916
Generation: 620. 正答率(トレーニング) = 0.91
 : 620. 正答率(テスト) = 0.9193
Generation: 630. 正答率(トレーニング) = 0.94
 : 630. 正答率(テスト) = 0.9217
Generation: 640. 正答率(トレーニング) = 0.96
 : 640. 正答率(テスト) = 0.9263
Generation: 650. 正答率(トレーニング) = 0.95
 : 650. 正答率(テスト) = 0.9294
Generation: 660. 正答率(トレーニング) = 0.89
 : 660. 正答率(テスト) = 0.9268
Generation: 670. 正答率(トレーニング) = 0.92
 : 670. 正答率(テスト) = 0.9307
Generation: 680. 正答率(トレーニング) = 0.96
 : 680. 正答率(テスト) = 0.9293
Generation: 690. 正答率(トレーニング) = 0.93

: 690. 正答率(テスト) = 0.9289
Generation: 700. 正答率(トレーニング) = 0.92
 : 700. 正答率(テスト) = 0.9305
Generation: 710. 正答率(トレーニング) = 0.94
 : 710. 正答率(テスト) = 0.9261
Generation: 720. 正答率(トレーニング) = 0.95
 : 720. 正答率(テスト) = 0.9277
Generation: 730. 正答率(トレーニング) = 0.96
 : 730. 正答率(テスト) = 0.9321
Generation: 740. 正答率(トレーニング) = 0.96
 : 740. 正答率(テスト) = 0.9324
Generation: 750. 正答率(トレーニング) = 0.96
 : 750. 正答率(テスト) = 0.9319
Generation: 760. 正答率(トレーニング) = 0.92
 : 760. 正答率(テスト) = 0.9312
Generation: 770. 正答率(トレーニング) = 0.95
 : 770. 正答率(テスト) = 0.9339
Generation: 780. 正答率(トレーニング) = 0.98
 : 780. 正答率(テスト) = 0.9358
Generation: 790. 正答率(トレーニング) = 0.92
 : 790. 正答率(テスト) = 0.9352
Generation: 800. 正答率(トレーニング) = 0.93
 : 800. 正答率(テスト) = 0.9319
Generation: 810. 正答率(トレーニング) = 0.97
 : 810. 正答率(テスト) = 0.9365
Generation: 820. 正答率(トレーニング) = 0.95
 : 820. 正答率(テスト) = 0.9274
Generation: 830. 正答率(トレーニング) = 0.93
 : 830. 正答率(テスト) = 0.9279
Generation: 840. 正答率(トレーニング) = 0.93
 : 840. 正答率(テスト) = 0.9354
Generation: 850. 正答率(トレーニング) = 0.89
 : 850. 正答率(テスト) = 0.9299
Generation: 860. 正答率(トレーニング) = 0.92
 : 860. 正答率(テスト) = 0.9358
Generation: 870. 正答率(トレーニング) = 0.97
 : 870. 正答率(テスト) = 0.9348
Generation: 880. 正答率(トレーニング) = 0.95
 : 880. 正答率(テスト) = 0.9348
Generation: 890. 正答率(トレーニング) = 0.97
 : 890. 正答率(テスト) = 0.9382
Generation: 900. 正答率(トレーニング) = 0.95

: 900. 正答率(テスト) = 0.9395
Generation: 910. 正答率(トレーニング) = 0.94
 : 910. 正答率(テスト) = 0.9352
Generation: 920. 正答率(トレーニング) = 0.94
 : 920. 正答率(テスト) = 0.9387
Generation: 930. 正答率(トレーニング) = 0.9
 : 930. 正答率(テスト) = 0.9359
Generation: 940. 正答率(トレーニング) = 0.93
 : 940. 正答率(テスト) = 0.9372
Generation: 950. 正答率(トレーニング) = 0.94
 : 950. 正答率(テスト) = 0.9393
Generation: 960. 正答率(トレーニング) = 0.95
 : 960. 正答率(テスト) = 0.9384
Generation: 970. 正答率(トレーニング) = 0.91
 : 970. 正答率(テスト) = 0.9342
Generation: 980. 正答率(トレーニング) = 0.95
 : 980. 正答率(テスト) = 0.9377
Generation: 990. 正答率(トレーニング) = 0.94
 : 990. 正答率(テスト) = 0.939
Generation: 1000. 正答率(トレーニング) = 0.97
 : 1000. 正答率(テスト) = 0.9374



[try] バッチ正規化をしてみよう & ReLU, He Momentum

```

In [48]: # データの読み込み
(x_train, d_train), (x_test, d_test) = load_mnist(normalize=True, one_hot_label=True)

print("データ読み込み完了")

# batch_normalizationの設定 =====
use_batchnorm = True
#use_batchnorm = False
# =====

network = MultiLayerNet(input_size=784, hidden_size_list=[40, 20], output_size=10, activation='relu', weight_init_std='he',
                        use_batchnorm=use_batchnorm)

iters_num = 1000
train_size = x_train.shape[0]
batch_size = 100
learning_rate = 0.01
# 慣性
momentum = 0.9

train_loss_list = []
accuracies_train = []
accuracies_test = []

plot_interval=10

for i in range(iters_num):
    idx_choice = np.random.choice(train_size, batch_size)
    x_batch = x_train[idx_choice]
    d_batch = d_train[idx_choice]

    # 勾配
    grad = network.gradient(x_batch, d_batch)
    if i == 0:
        v = {}
    for key in ('W1', 'W2', 'W3', 'b1', 'b2', 'b3'):
        if i == 0:
            v[key] = np.zeros_like(network.params[key])
            print("i=0", key, "v=", v[key])
        v[key] = momentum * v[key] - learning_rate * grad[key]
        network.params[key] += v[key]

```



```
    loss = network.loss(x_batch, d_batch)
    train_loss_list.append(loss)

    if (i + 1) % plot_interval == 0:
        accr_test = network.accuracy(x_test, d_test)
        accuracies_test.append(accr_test)
        accr_train = network.accuracy(x_batch, d_batch)
        accuracies_train.append(accr_train)

        print('Generation: ' + str(i+1) + '. 正答率(トレーニング) = ' + str(accr_train))
        print('                : ' + str(i+1) + '. 正答率(テスト) = ' + str(accr_test))

lists = range(0, iters_num, plot_interval)
plt.plot(lists, accuracies_train, label="training set")
plt.plot(lists, accuracies_test, label="test set")
plt.legend(loc="lower right")
plt.title("accuracy")
plt.xlabel("count")
plt.ylabel("accuracy")
plt.ylim(0, 1.0)
# グラフの表示
plt.show()
```

[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

file:///C:/Users/克扩/Desktop/upload/2_4_optimizer-ensyu.html

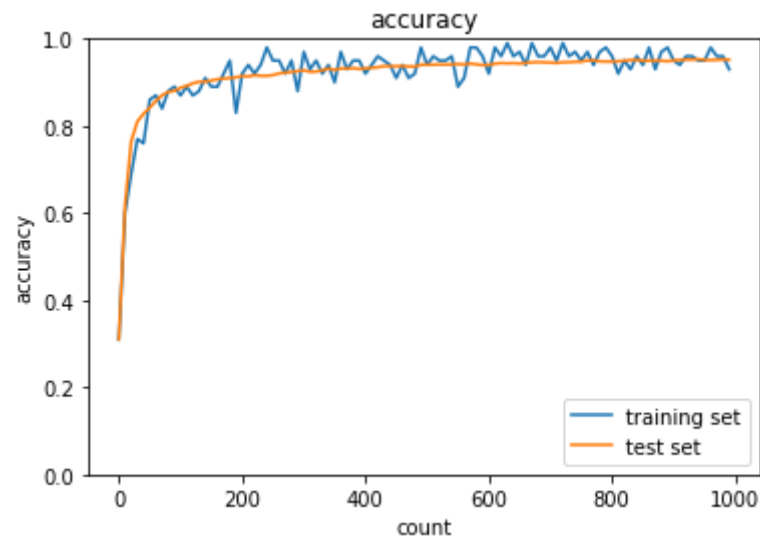
: 60. 正答率(テスト) = 0.8419
Generation: 70. 正答率(トレーニング) = 0.87
 : 70. 正答率(テスト) = 0.8573
Generation: 80. 正答率(トレーニング) = 0.84
 : 80. 正答率(テスト) = 0.8704
Generation: 90. 正答率(トレーニング) = 0.88
 : 90. 正答率(テスト) = 0.878
Generation: 100. 正答率(トレーニング) = 0.89
 : 100. 正答率(テスト) = 0.8805
Generation: 110. 正答率(トレーニング) = 0.87
 : 110. 正答率(テスト) = 0.8876
Generation: 120. 正答率(トレーニング) = 0.89
 : 120. 正答率(テスト) = 0.8919
Generation: 130. 正答率(トレーニング) = 0.87
 : 130. 正答率(テスト) = 0.8979
Generation: 140. 正答率(トレーニング) = 0.88
 : 140. 正答率(テスト) = 0.9013
Generation: 150. 正答率(トレーニング) = 0.91
 : 150. 正答率(テスト) = 0.9017
Generation: 160. 正答率(トレーニング) = 0.89
 : 160. 正答率(テスト) = 0.9043
Generation: 170. 正答率(トレーニング) = 0.89
 : 170. 正答率(テスト) = 0.9071
Generation: 180. 正答率(トレーニング) = 0.92
 : 180. 正答率(テスト) = 0.9088
Generation: 190. 正答率(トレーニング) = 0.95
 : 190. 正答率(テスト) = 0.9097
Generation: 200. 正答率(トレーニング) = 0.83
 : 200. 正答率(テスト) = 0.9116
Generation: 210. 正答率(トレーニング) = 0.92
 : 210. 正答率(テスト) = 0.9147
Generation: 220. 正答率(トレーニング) = 0.94
 : 220. 正答率(テスト) = 0.9135
Generation: 230. 正答率(トレーニング) = 0.92
 : 230. 正答率(テスト) = 0.9165
Generation: 240. 正答率(トレーニング) = 0.94
 : 240. 正答率(テスト) = 0.9152
Generation: 250. 正答率(トレーニング) = 0.98
 : 250. 正答率(テスト) = 0.9149
Generation: 260. 正答率(トレーニング) = 0.95
 : 260. 正答率(テスト) = 0.9168
Generation: 270. 正答率(トレーニング) = 0.95

: 270. 正答率(テスト) = 0.9207
Generation: 280. 正答率(トレーニング) = 0.92
 : 280. 正答率(テスト) = 0.9223
Generation: 290. 正答率(トレーニング) = 0.95
 : 290. 正答率(テスト) = 0.9237
Generation: 300. 正答率(トレーニング) = 0.88
 : 300. 正答率(テスト) = 0.9259
Generation: 310. 正答率(トレーニング) = 0.97
 : 310. 正答率(テスト) = 0.9277
Generation: 320. 正答率(トレーニング) = 0.93
 : 320. 正答率(テスト) = 0.9242
Generation: 330. 正答率(トレーニング) = 0.95
 : 330. 正答率(テスト) = 0.9248
Generation: 340. 正答率(トレーニング) = 0.92
 : 340. 正答率(テスト) = 0.9279
Generation: 350. 正答率(トレーニング) = 0.94
 : 350. 正答率(テスト) = 0.9316
Generation: 360. 正答率(トレーニング) = 0.9
 : 360. 正答率(テスト) = 0.9286
Generation: 370. 正答率(トレーニング) = 0.97
 : 370. 正答率(テスト) = 0.9316
Generation: 380. 正答率(トレーニング) = 0.93
 : 380. 正答率(テスト) = 0.9315
Generation: 390. 正答率(トレーニング) = 0.95
 : 390. 正答率(テスト) = 0.9327
Generation: 400. 正答率(トレーニング) = 0.95
 : 400. 正答率(テスト) = 0.9306
Generation: 410. 正答率(トレーニング) = 0.92
 : 410. 正答率(テスト) = 0.9319
Generation: 420. 正答率(トレーニング) = 0.94
 : 420. 正答率(テスト) = 0.9333
Generation: 430. 正答率(トレーニング) = 0.96
 : 430. 正答率(テスト) = 0.9347
Generation: 440. 正答率(トレーニング) = 0.95
 : 440. 正答率(テスト) = 0.9373
Generation: 450. 正答率(トレーニング) = 0.94
 : 450. 正答率(テスト) = 0.9369
Generation: 460. 正答率(トレーニング) = 0.91
 : 460. 正答率(テスト) = 0.937
Generation: 470. 正答率(トレーニング) = 0.94
 : 470. 正答率(テスト) = 0.9376
Generation: 480. 正答率(トレーニング) = 0.91

: 480. 正答率(テスト) = 0.9366
Generation: 490. 正答率(トレーニング) = 0.92
 : 490. 正答率(テスト) = 0.9362
Generation: 500. 正答率(トレーニング) = 0.98
 : 500. 正答率(テスト) = 0.9404
Generation: 510. 正答率(トレーニング) = 0.94
 : 510. 正答率(テスト) = 0.9405
Generation: 520. 正答率(トレーニング) = 0.96
 : 520. 正答率(テスト) = 0.9404
Generation: 530. 正答率(トレーニング) = 0.95
 : 530. 正答率(テスト) = 0.9412
Generation: 540. 正答率(トレーニング) = 0.95
 : 540. 正答率(テスト) = 0.9409
Generation: 550. 正答率(トレーニング) = 0.96
 : 550. 正答率(テスト) = 0.941
Generation: 560. 正答率(トレーニング) = 0.89
 : 560. 正答率(テスト) = 0.9421
Generation: 570. 正答率(トレーニング) = 0.91
 : 570. 正答率(テスト) = 0.941
Generation: 580. 正答率(トレーニング) = 0.98
 : 580. 正答率(テスト) = 0.9431
Generation: 590. 正答率(トレーニング) = 0.98
 : 590. 正答率(テスト) = 0.9414
Generation: 600. 正答率(トレーニング) = 0.96
 : 600. 正答率(テスト) = 0.9395
Generation: 610. 正答率(トレーニング) = 0.92
 : 610. 正答率(テスト) = 0.9397
Generation: 620. 正答率(トレーニング) = 0.98
 : 620. 正答率(テスト) = 0.9429
Generation: 630. 正答率(トレーニング) = 0.96
 : 630. 正答率(テスト) = 0.9443
Generation: 640. 正答率(トレーニング) = 0.99
 : 640. 正答率(テスト) = 0.9436
Generation: 650. 正答率(トレーニング) = 0.96
 : 650. 正答率(テスト) = 0.9442
Generation: 660. 正答率(トレーニング) = 0.97
 : 660. 正答率(テスト) = 0.9431
Generation: 670. 正答率(トレーニング) = 0.94
 : 670. 正答率(テスト) = 0.9445
Generation: 680. 正答率(トレーニング) = 0.99
 : 680. 正答率(テスト) = 0.9459
Generation: 690. 正答率(トレーニング) = 0.96

: 690. 正答率(テスト) = 0.9465
Generation: 700. 正答率(トレーニング) = 0.96
 : 700. 正答率(テスト) = 0.9459
Generation: 710. 正答率(トレーニング) = 0.98
 : 710. 正答率(テスト) = 0.9453
Generation: 720. 正答率(トレーニング) = 0.95
 : 720. 正答率(テスト) = 0.9445
Generation: 730. 正答率(トレーニング) = 0.99
 : 730. 正答率(テスト) = 0.946
Generation: 740. 正答率(トレーニング) = 0.96
 : 740. 正答率(テスト) = 0.9463
Generation: 750. 正答率(トレーニング) = 0.97
 : 750. 正答率(テスト) = 0.9472
Generation: 760. 正答率(トレーニング) = 0.95
 : 760. 正答率(テスト) = 0.9475
Generation: 770. 正答率(トレーニング) = 0.97
 : 770. 正答率(テスト) = 0.9502
Generation: 780. 正答率(トレーニング) = 0.94
 : 780. 正答率(テスト) = 0.9501
Generation: 790. 正答率(トレーニング) = 0.97
 : 790. 正答率(テスト) = 0.9484
Generation: 800. 正答率(トレーニング) = 0.98
 : 800. 正答率(テスト) = 0.9479
Generation: 810. 正答率(トレーニング) = 0.96
 : 810. 正答率(テスト) = 0.948
Generation: 820. 正答率(トレーニング) = 0.92
 : 820. 正答率(テスト) = 0.9485
Generation: 830. 正答率(トレーニング) = 0.95
 : 830. 正答率(テスト) = 0.9496
Generation: 840. 正答率(トレーニング) = 0.93
 : 840. 正答率(テスト) = 0.9511
Generation: 850. 正答率(トレーニング) = 0.96
 : 850. 正答率(テスト) = 0.9515
Generation: 860. 正答率(トレーニング) = 0.94
 : 860. 正答率(テスト) = 0.9495
Generation: 870. 正答率(トレーニング) = 0.98
 : 870. 正答率(テスト) = 0.9495
Generation: 880. 正答率(トレーニング) = 0.93
 : 880. 正答率(テスト) = 0.9497
Generation: 890. 正答率(トレーニング) = 0.97
 : 890. 正答率(テスト) = 0.9496
Generation: 900. 正答率(トレーニング) = 0.98

: 900. 正答率(テスト) = 0.9486
Generation: 910. 正答率(トレーニング) = 0.95
 : 910. 正答率(テスト) = 0.9501
Generation: 920. 正答率(トレーニング) = 0.94
 : 920. 正答率(テスト) = 0.9514
Generation: 930. 正答率(トレーニング) = 0.96
 : 930. 正答率(テスト) = 0.9525
Generation: 940. 正答率(トレーニング) = 0.96
 : 940. 正答率(テスト) = 0.9523
Generation: 950. 正答率(トレーニング) = 0.95
 : 950. 正答率(テスト) = 0.9533
Generation: 960. 正答率(トレーニング) = 0.95
 : 960. 正答率(テスト) = 0.9517
Generation: 970. 正答率(トレーニング) = 0.98
 : 970. 正答率(テスト) = 0.951
Generation: 980. 正答率(トレーニング) = 0.96
 : 980. 正答率(テスト) = 0.9519
Generation: 990. 正答率(トレーニング) = 0.96
 : 990. 正答率(テスト) = 0.9525
Generation: 1000. 正答率(トレーニング) = 0.93
 : 1000. 正答率(テスト) = 0.9521



MomentumをもとにAdaGradを作ってみよう

$\theta = 1e-4$ とする

```
In [21]: # AdaGradを作ってみよう
# データの読み込み
(x_train, d_train), (x_test, d_test) = load_mnist(normalize=True, one_hot_label=True)

print("データ読み込み完了")

# batch_normalizationの設定 =====
#use_batchnorm = True
use_batchnorm = False
# =====

network = MultiLayerNet(input_size=784, hidden_size_list=[40, 20], output_size=10, activation='sigmoid', weight_init_std=0.01,
                        use_batchnorm=use_batchnorm)

iters_num = 1000
# iters_num = 500 # 処理を短縮

train_size = x_train.shape[0]
batch_size = 100
learning_rate = 0.01

# AdaGradでは不必要
# =====

#momentum = 0.9

# =====

train_loss_list = []
accuracies_train = []
accuracies_test = []

plot_interval=10

for i in range(iters_num):
    batch_mask = np.random.choice(train_size, batch_size)
    x_batch = x_train[batch_mask]
    d_batch = d_train[batch_mask]

    # 勾配
    grad = network.gradient(x_batch, d_batch)
```

```

if i == 0:
    h = {}
for key in ('W1', 'W2', 'W3', 'b1', 'b2', 'b3'):

    # 変更しよう
    # =====
    if i == 0:
        h[key] = np.zeros_like(network.params[key])+1e-4
        h[key] = h[key] + grad[key] * grad[key]
        network.params[key] -= learning_rate * grad[key] / (np.sqrt(h[key])+ 1e-4 )

    # =====

    loss = network.loss(x_batch, d_batch)
    train_loss_list.append(loss)

if (i + 1) % plot_interval == 0:
    accr_test = network.accuracy(x_test, d_test)
    accuracies_test.append(accr_test)
    accr_train = network.accuracy(x_batch, d_batch)
    accuracies_train.append(accr_train)

    print('Generation: ' + str(i+1) + '. 正答率(トレーニング) = ' + str(accr_train))
    print('          : ' + str(i+1) + '. 正答率(テスト) = ' + str(accr_test))

lists = range(0, iters_num, plot_interval)
plt.plot(lists, accuracies_train, label="training set")
plt.plot(lists, accuracies_test, label="test set")
plt.legend(loc="lower right")
plt.title("accuracy")
plt.xlabel("count")
plt.ylabel("accuracy")
plt.ylim(0, 1.0)
# グラフの表示
plt.show()

```

データ読み込み完了

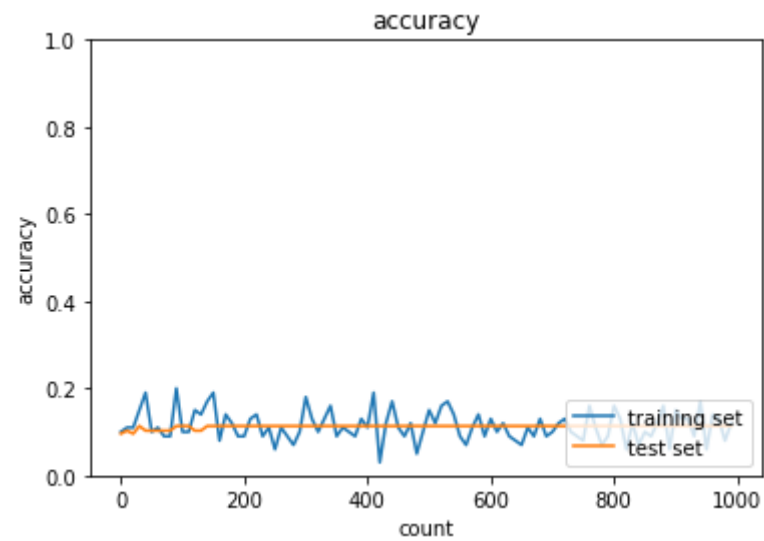
Generation: 10. 正答率(トレーニング) = 0.1
: 10. 正答率(テスト) = 0.0958
Generation: 20. 正答率(トレーニング) = 0.11
: 20. 正答率(テスト) = 0.1028
Generation: 30. 正答率(トレーニング) = 0.11
: 30. 正答率(テスト) = 0.0958
Generation: 40. 正答率(トレーニング) = 0.15
: 40. 正答率(テスト) = 0.1135
Generation: 50. 正答率(トレーニング) = 0.19
: 50. 正答率(テスト) = 0.1028
Generation: 60. 正答率(トレーニング) = 0.1
: 60. 正答率(テスト) = 0.1028
Generation: 70. 正答率(トレーニング) = 0.11
: 70. 正答率(テスト) = 0.1028
Generation: 80. 正答率(トレーニング) = 0.09
: 80. 正答率(テスト) = 0.1028
Generation: 90. 正答率(トレーニング) = 0.09
: 90. 正答率(テスト) = 0.1028
Generation: 100. 正答率(トレーニング) = 0.2
: 100. 正答率(テスト) = 0.1135
Generation: 110. 正答率(トレーニング) = 0.1
: 110. 正答率(テスト) = 0.1135
Generation: 120. 正答率(トレーニング) = 0.1
: 120. 正答率(テスト) = 0.1135
Generation: 130. 正答率(トレーニング) = 0.15
: 130. 正答率(テスト) = 0.1028
Generation: 140. 正答率(トレーニング) = 0.14
: 140. 正答率(テスト) = 0.1028
Generation: 150. 正答率(トレーニング) = 0.17
: 150. 正答率(テスト) = 0.1135
Generation: 160. 正答率(トレーニング) = 0.19
: 160. 正答率(テスト) = 0.1135
Generation: 170. 正答率(トレーニング) = 0.08
: 170. 正答率(テスト) = 0.1135
Generation: 180. 正答率(トレーニング) = 0.14
: 180. 正答率(テスト) = 0.1135
Generation: 190. 正答率(トレーニング) = 0.12
: 190. 正答率(テスト) = 0.1135
Generation: 200. 正答率(トレーニング) = 0.09
: 200. 正答率(テスト) = 0.1135

Generation: 210. 正答率(トレーニング) = 0.09
: 210. 正答率(テスト) = 0.1135
Generation: 220. 正答率(トレーニング) = 0.13
: 220. 正答率(テスト) = 0.1135
Generation: 230. 正答率(トレーニング) = 0.14
: 230. 正答率(テスト) = 0.1135
Generation: 240. 正答率(トレーニング) = 0.09
: 240. 正答率(テスト) = 0.1135
Generation: 250. 正答率(トレーニング) = 0.11
: 250. 正答率(テスト) = 0.1135
Generation: 260. 正答率(トレーニング) = 0.06
: 260. 正答率(テスト) = 0.1135
Generation: 270. 正答率(トレーニング) = 0.11
: 270. 正答率(テスト) = 0.1135
Generation: 280. 正答率(トレーニング) = 0.09
: 280. 正答率(テスト) = 0.1135
Generation: 290. 正答率(トレーニング) = 0.07
: 290. 正答率(テスト) = 0.1135
Generation: 300. 正答率(トレーニング) = 0.1
: 300. 正答率(テスト) = 0.1135
Generation: 310. 正答率(トレーニング) = 0.18
: 310. 正答率(テスト) = 0.1135
Generation: 320. 正答率(トレーニング) = 0.13
: 320. 正答率(テスト) = 0.1135
Generation: 330. 正答率(トレーニング) = 0.1
: 330. 正答率(テスト) = 0.1135
Generation: 340. 正答率(トレーニング) = 0.13
: 340. 正答率(テスト) = 0.1135
Generation: 350. 正答率(トレーニング) = 0.16
: 350. 正答率(テスト) = 0.1135
Generation: 360. 正答率(トレーニング) = 0.09
: 360. 正答率(テスト) = 0.1135
Generation: 370. 正答率(トレーニング) = 0.11
: 370. 正答率(テスト) = 0.1135
Generation: 380. 正答率(トレーニング) = 0.1
: 380. 正答率(テスト) = 0.1135
Generation: 390. 正答率(トレーニング) = 0.09
: 390. 正答率(テスト) = 0.1135
Generation: 400. 正答率(トレーニング) = 0.13
: 400. 正答率(テスト) = 0.1135
Generation: 410. 正答率(トレーニング) = 0.11
: 410. 正答率(テスト) = 0.1135

Generation: 420. 正答率(トレーニング) = 0.19
: 420. 正答率(テスト) = 0.1135
Generation: 430. 正答率(トレーニング) = 0.03
: 430. 正答率(テスト) = 0.1135
Generation: 440. 正答率(トレーニング) = 0.12
: 440. 正答率(テスト) = 0.1135
Generation: 450. 正答率(トレーニング) = 0.17
: 450. 正答率(テスト) = 0.1135
Generation: 460. 正答率(トレーニング) = 0.11
: 460. 正答率(テスト) = 0.1135
Generation: 470. 正答率(トレーニング) = 0.09
: 470. 正答率(テスト) = 0.1135
Generation: 480. 正答率(トレーニング) = 0.12
: 480. 正答率(テスト) = 0.1135
Generation: 490. 正答率(トレーニング) = 0.05
: 490. 正答率(テスト) = 0.1135
Generation: 500. 正答率(トレーニング) = 0.1
: 500. 正答率(テスト) = 0.1135
Generation: 510. 正答率(トレーニング) = 0.15
: 510. 正答率(テスト) = 0.1135
Generation: 520. 正答率(トレーニング) = 0.12
: 520. 正答率(テスト) = 0.1135
Generation: 530. 正答率(トレーニング) = 0.16
: 530. 正答率(テスト) = 0.1135
Generation: 540. 正答率(トレーニング) = 0.17
: 540. 正答率(テスト) = 0.1135
Generation: 550. 正答率(トレーニング) = 0.14
: 550. 正答率(テスト) = 0.1135
Generation: 560. 正答率(トレーニング) = 0.09
: 560. 正答率(テスト) = 0.1135
Generation: 570. 正答率(トレーニング) = 0.07
: 570. 正答率(テスト) = 0.1135
Generation: 580. 正答率(トレーニング) = 0.11
: 580. 正答率(テスト) = 0.1135
Generation: 590. 正答率(トレーニング) = 0.14
: 590. 正答率(テスト) = 0.1135
Generation: 600. 正答率(トレーニング) = 0.09
: 600. 正答率(テスト) = 0.1135
Generation: 610. 正答率(トレーニング) = 0.13
: 610. 正答率(テスト) = 0.1135
Generation: 620. 正答率(トレーニング) = 0.1
: 620. 正答率(テスト) = 0.1135

Generation: 630. 正答率(トレーニング) = 0.12
: 630. 正答率(テスト) = 0.1135
Generation: 640. 正答率(トレーニング) = 0.09
: 640. 正答率(テスト) = 0.1135
Generation: 650. 正答率(トレーニング) = 0.08
: 650. 正答率(テスト) = 0.1135
Generation: 660. 正答率(トレーニング) = 0.07
: 660. 正答率(テスト) = 0.1135
Generation: 670. 正答率(トレーニング) = 0.11
: 670. 正答率(テスト) = 0.1135
Generation: 680. 正答率(トレーニング) = 0.09
: 680. 正答率(テスト) = 0.1135
Generation: 690. 正答率(トレーニング) = 0.13
: 690. 正答率(テスト) = 0.1135
Generation: 700. 正答率(トレーニング) = 0.09
: 700. 正答率(テスト) = 0.1135
Generation: 710. 正答率(トレーニング) = 0.1
: 710. 正答率(テスト) = 0.1135
Generation: 720. 正答率(トレーニング) = 0.12
: 720. 正答率(テスト) = 0.1135
Generation: 730. 正答率(トレーニング) = 0.13
: 730. 正答率(テスト) = 0.1135
Generation: 740. 正答率(トレーニング) = 0.1
: 740. 正答率(テスト) = 0.1135
Generation: 750. 正答率(トレーニング) = 0.09
: 750. 正答率(テスト) = 0.1135
Generation: 760. 正答率(トレーニング) = 0.08
: 760. 正答率(テスト) = 0.1135
Generation: 770. 正答率(トレーニング) = 0.16
: 770. 正答率(テスト) = 0.1135
Generation: 780. 正答率(トレーニング) = 0.11
: 780. 正答率(テスト) = 0.1135
Generation: 790. 正答率(トレーニング) = 0.07
: 790. 正答率(テスト) = 0.1135
Generation: 800. 正答率(トレーニング) = 0.09
: 800. 正答率(テスト) = 0.1135
Generation: 810. 正答率(トレーニング) = 0.16
: 810. 正答率(テスト) = 0.1135
Generation: 820. 正答率(トレーニング) = 0.13
: 820. 正答率(テスト) = 0.1135
Generation: 830. 正答率(トレーニング) = 0.06
: 830. 正答率(テスト) = 0.1135

Generation: 840. 正答率(トレーニング) = 0.12
: 840. 正答率(テスト) = 0.1135
Generation: 850. 正答率(トレーニング) = 0.07
: 850. 正答率(テスト) = 0.1135
Generation: 860. 正答率(トレーニング) = 0.1
: 860. 正答率(テスト) = 0.1135
Generation: 870. 正答率(トレーニング) = 0.09
: 870. 正答率(テスト) = 0.1135
Generation: 880. 正答率(トレーニング) = 0.12
: 880. 正答率(テスト) = 0.1135
Generation: 890. 正答率(トレーニング) = 0.16
: 890. 正答率(テスト) = 0.1135
Generation: 900. 正答率(トレーニング) = 0.06
: 900. 正答率(テスト) = 0.1135
Generation: 910. 正答率(トレーニング) = 0.15
: 910. 正答率(テスト) = 0.1135
Generation: 920. 正答率(トレーニング) = 0.13
: 920. 正答率(テスト) = 0.1135
Generation: 930. 正答率(トレーニング) = 0.13
: 930. 正答率(テスト) = 0.1135
Generation: 940. 正答率(トレーニング) = 0.09
: 940. 正答率(テスト) = 0.1135
Generation: 950. 正答率(トレーニング) = 0.17
: 950. 正答率(テスト) = 0.1135
Generation: 960. 正答率(トレーニング) = 0.06
: 960. 正答率(テスト) = 0.1135
Generation: 970. 正答率(トレーニング) = 0.14
: 970. 正答率(テスト) = 0.1135
Generation: 980. 正答率(トレーニング) = 0.13
: 980. 正答率(テスト) = 0.1135
Generation: 990. 正答率(トレーニング) = 0.08
: 990. 正答率(テスト) = 0.1135
Generation: 1000. 正答率(トレーニング) = 0.12
: 1000. 正答率(テスト) = 0.1135



[try] 学習率を変えてみよう

AdaGrad

```
In [23]: # AdaGradを作ってみよう
# データの読み込み
(x_train, d_train), (x_test, d_test) = load_mnist(normalize=True, one_hot_label=True)

print("データ読み込み完了")

# batch_normalizationの設定 =====
#use_batchnorm = True
use_batchnorm = False
# =====

network = MultiLayerNet(input_size=784, hidden_size_list=[40, 20], output_size=10, activation='sigmoid', weight_init_std=0.01,
                        use_batchnorm=use_batchnorm)

iters_num = 1000
# iters_num = 500 # 処理を短縮

train_size = x_train.shape[0]
batch_size = 100
learning_rate = 100

# AdaGradでは不必要
# =====

#momentum = 0.9

# =====

train_loss_list = []
accuracies_train = []
accuracies_test = []

plot_interval=10

for i in range(iters_num):
    batch_mask = np.random.choice(train_size, batch_size)
    x_batch = x_train[batch_mask]
    d_batch = d_train[batch_mask]

    # 勾配
    grad = network.gradient(x_batch, d_batch)
```

```

if i == 0:
    h = {}
for key in ('W1', 'W2', 'W3', 'b1', 'b2', 'b3'):

    # 変更しよう
    # =====
    if i == 0:
        h[key] = np.zeros_like(network.params[key])+1e-4
        h[key] = h[key] + grad[key] * grad[key]
        network.params[key] -= learning_rate * grad[key] / (np.sqrt(h[key])+ 1e-4 )

    # =====

    loss = network.loss(x_batch, d_batch)
    train_loss_list.append(loss)

if (i + 1) % plot_interval == 0:
    accr_test = network.accuracy(x_test, d_test)
    accuracies_test.append(accr_test)
    accr_train = network.accuracy(x_batch, d_batch)
    accuracies_train.append(accr_train)

    print('Generation: ' + str(i+1) + '. 正答率(トレーニング) = ' + str(accr_train))
    print('          : ' + str(i+1) + '. 正答率(テスト) = ' + str(accr_test))

lists = range(0, iters_num, plot_interval)
plt.plot(lists, accuracies_train, label="training set")
plt.plot(lists, accuracies_test, label="test set")
plt.legend(loc="lower right")
plt.title("accuracy")
plt.xlabel("count")
plt.ylabel("accuracy")
plt.ylim(0, 1.0)
# グラフの表示
plt.show()

```

データ読み込み完了

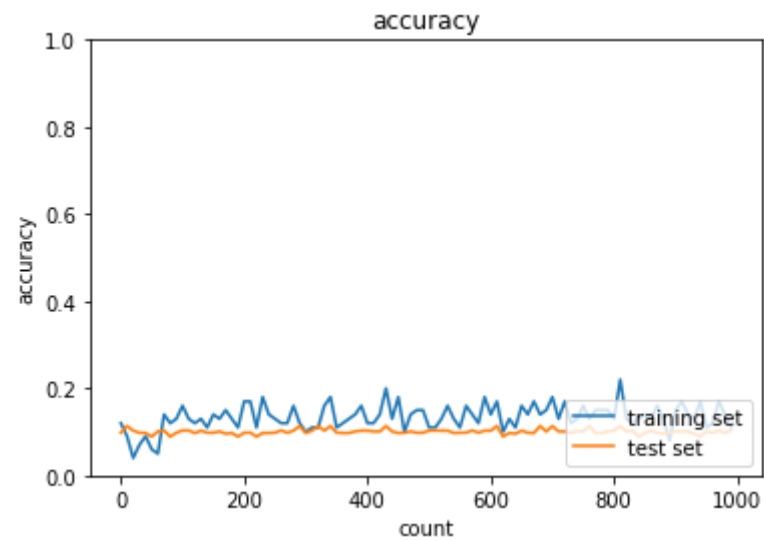
Generation: 10. 正答率(トレーニング) = 0.12
: 10. 正答率(テスト) = 0.0982
Generation: 20. 正答率(トレーニング) = 0.09
: 20. 正答率(テスト) = 0.1135
Generation: 30. 正答率(トレーニング) = 0.04
: 30. 正答率(テスト) = 0.1028
Generation: 40. 正答率(トレーニング) = 0.07
: 40. 正答率(テスト) = 0.098
Generation: 50. 正答率(トレーニング) = 0.09
: 50. 正答率(テスト) = 0.0974
Generation: 60. 正答率(トレーニング) = 0.06
: 60. 正答率(テスト) = 0.0892
Generation: 70. 正答率(トレーニング) = 0.05
: 70. 正答率(テスト) = 0.1009
Generation: 80. 正答率(トレーニング) = 0.14
: 80. 正答率(テスト) = 0.1032
Generation: 90. 正答率(トレーニング) = 0.12
: 90. 正答率(テスト) = 0.0892
Generation: 100. 正答率(トレーニング) = 0.13
: 100. 正答率(テスト) = 0.098
Generation: 110. 正答率(トレーニング) = 0.16
: 110. 正答率(テスト) = 0.1032
Generation: 120. 正答率(トレーニング) = 0.13
: 120. 正答率(テスト) = 0.1032
Generation: 130. 正答率(トレーニング) = 0.12
: 130. 正答率(テスト) = 0.0974
Generation: 140. 正答率(トレーニング) = 0.13
: 140. 正答率(テスト) = 0.1028
Generation: 150. 正答率(トレーニング) = 0.11
: 150. 正答率(テスト) = 0.0982
Generation: 160. 正答率(トレーニング) = 0.14
: 160. 正答率(テスト) = 0.098
Generation: 170. 正答率(トレーニング) = 0.13
: 170. 正答率(テスト) = 0.101
Generation: 180. 正答率(トレーニング) = 0.15
: 180. 正答率(テスト) = 0.0958
Generation: 190. 正答率(トレーニング) = 0.13
: 190. 正答率(テスト) = 0.0974
Generation: 200. 正答率(トレーニング) = 0.11
: 200. 正答率(テスト) = 0.0892

Generation: 210. 正答率(トレーニング) = 0.17
: 210. 正答率(テスト) = 0.098
Generation: 220. 正答率(トレーニング) = 0.17
: 220. 正答率(テスト) = 0.0982
Generation: 230. 正答率(トレーニング) = 0.11
: 230. 正答率(テスト) = 0.0892
Generation: 240. 正答率(トレーニング) = 0.18
: 240. 正答率(テスト) = 0.0974
Generation: 250. 正答率(トレーニング) = 0.14
: 250. 正答率(テスト) = 0.0974
Generation: 260. 正答率(トレーニング) = 0.13
: 260. 正答率(テスト) = 0.0982
Generation: 270. 正答率(トレーニング) = 0.12
: 270. 正答率(テスト) = 0.1032
Generation: 280. 正答率(トレーニング) = 0.12
: 280. 正答率(テスト) = 0.098
Generation: 290. 正答率(トレーニング) = 0.16
: 290. 正答率(テスト) = 0.1028
Generation: 300. 正答率(トレーニング) = 0.12
: 300. 正答率(テスト) = 0.1135
Generation: 310. 正答率(トレーニング) = 0.1
: 310. 正答率(テスト) = 0.0974
Generation: 320. 正答率(トレーニング) = 0.11
: 320. 正答率(テスト) = 0.1028
Generation: 330. 正答率(トレーニング) = 0.11
: 330. 正答率(テスト) = 0.1135
Generation: 340. 正答率(トレーニング) = 0.16
: 340. 正答率(テスト) = 0.1028
Generation: 350. 正答率(トレーニング) = 0.18
: 350. 正答率(テスト) = 0.1135
Generation: 360. 正答率(トレーニング) = 0.11
: 360. 正答率(テスト) = 0.0982
Generation: 370. 正答率(トレーニング) = 0.12
: 370. 正答率(テスト) = 0.0974
Generation: 380. 正答率(トレーニング) = 0.13
: 380. 正答率(テスト) = 0.0974
Generation: 390. 正答率(トレーニング) = 0.14
: 390. 正答率(テスト) = 0.101
Generation: 400. 正答率(トレーニング) = 0.16
: 400. 正答率(テスト) = 0.1028
Generation: 410. 正答率(トレーニング) = 0.12
: 410. 正答率(テスト) = 0.1028

Generation: 420. 正答率(トレーニング) = 0.12
: 420. 正答率(テスト) = 0.101
Generation: 430. 正答率(トレーニング) = 0.14
: 430. 正答率(テスト) = 0.1009
Generation: 440. 正答率(トレーニング) = 0.2
: 440. 正答率(テスト) = 0.1135
Generation: 450. 正答率(トレーニング) = 0.13
: 450. 正答率(テスト) = 0.1009
Generation: 460. 正答率(トレーニング) = 0.18
: 460. 正答率(テスト) = 0.0974
Generation: 470. 正答率(トレーニング) = 0.1
: 470. 正答率(テスト) = 0.098
Generation: 480. 正答率(トレーニング) = 0.14
: 480. 正答率(テスト) = 0.101
Generation: 490. 正答率(トレーニング) = 0.15
: 490. 正答率(テスト) = 0.098
Generation: 500. 正答率(トレーニング) = 0.15
: 500. 正答率(テスト) = 0.0982
Generation: 510. 正答率(トレーニング) = 0.11
: 510. 正答率(テスト) = 0.1028
Generation: 520. 正答率(トレーニング) = 0.11
: 520. 正答率(テスト) = 0.1032
Generation: 530. 正答率(トレーニング) = 0.13
: 530. 正答率(テスト) = 0.1028
Generation: 540. 正答率(トレーニング) = 0.16
: 540. 正答率(テスト) = 0.1028
Generation: 550. 正答率(トレーニング) = 0.13
: 550. 正答率(テスト) = 0.0974
Generation: 560. 正答率(トレーニング) = 0.11
: 560. 正答率(テスト) = 0.0982
Generation: 570. 正答率(トレーニング) = 0.16
: 570. 正答率(テスト) = 0.0982
Generation: 580. 正答率(トレーニング) = 0.14
: 580. 正答率(テスト) = 0.1032
Generation: 590. 正答率(トレーニング) = 0.12
: 590. 正答率(テスト) = 0.0982
Generation: 600. 正答率(トレーニング) = 0.18
: 600. 正答率(テスト) = 0.1028
Generation: 610. 正答率(トレーニング) = 0.14
: 610. 正答率(テスト) = 0.1028
Generation: 620. 正答率(トレーニング) = 0.17
: 620. 正答率(テスト) = 0.1135

Generation: 630. 正答率(トレーニング) = 0.1
: 630. 正答率(テスト) = 0.0892
Generation: 640. 正答率(トレーニング) = 0.13
: 640. 正答率(テスト) = 0.0974
Generation: 650. 正答率(トレーニング) = 0.11
: 650. 正答率(テスト) = 0.0958
Generation: 660. 正答率(トレーニング) = 0.16
: 660. 正答率(テスト) = 0.1028
Generation: 670. 正答率(トレーニング) = 0.14
: 670. 正答率(テスト) = 0.098
Generation: 680. 正答率(トレーニング) = 0.17
: 680. 正答率(テスト) = 0.0974
Generation: 690. 正答率(トレーニング) = 0.14
: 690. 正答率(テスト) = 0.1135
Generation: 700. 正答率(トレーニング) = 0.15
: 700. 正答率(テスト) = 0.1009
Generation: 710. 正答率(トレーニング) = 0.18
: 710. 正答率(テスト) = 0.1135
Generation: 720. 正答率(トレーニング) = 0.13
: 720. 正答率(テスト) = 0.101
Generation: 730. 正答率(トレーニング) = 0.17
: 730. 正答率(テスト) = 0.1009
Generation: 740. 正答率(トレーニング) = 0.12
: 740. 正答率(テスト) = 0.0982
Generation: 750. 正答率(トレーニング) = 0.13
: 750. 正答率(テスト) = 0.1009
Generation: 760. 正答率(トレーニング) = 0.16
: 760. 正答率(テスト) = 0.1009
Generation: 770. 正答率(トレーニング) = 0.13
: 770. 正答率(テスト) = 0.1135
Generation: 780. 正答率(トレーニング) = 0.15
: 780. 正答率(テスト) = 0.0974
Generation: 790. 正答率(トレーニング) = 0.15
: 790. 正答率(テスト) = 0.0974
Generation: 800. 正答率(トレーニング) = 0.15
: 800. 正答率(テスト) = 0.101
Generation: 810. 正答率(トレーニング) = 0.13
: 810. 正答率(テスト) = 0.1028
Generation: 820. 正答率(トレーニング) = 0.22
: 820. 正答率(テスト) = 0.1135
Generation: 830. 正答率(トレーニング) = 0.13
: 830. 正答率(テスト) = 0.101

Generation: 840. 正答率(トレーニング) = 0.12
: 840. 正答率(テスト) = 0.1032
Generation: 850. 正答率(トレーニング) = 0.13
: 850. 正答率(テスト) = 0.0892
Generation: 860. 正答率(トレーニング) = 0.14
: 860. 正答率(テスト) = 0.0974
Generation: 870. 正答率(トレーニング) = 0.13
: 870. 正答率(テスト) = 0.1028
Generation: 880. 正答率(トレーニング) = 0.16
: 880. 正答率(テスト) = 0.098
Generation: 890. 正答率(トレーニング) = 0.13
: 890. 正答率(テスト) = 0.0958
Generation: 900. 正答率(トレーニング) = 0.08
: 900. 正答率(テスト) = 0.0982
Generation: 910. 正答率(トレーニング) = 0.15
: 910. 正答率(テスト) = 0.101
Generation: 920. 正答率(トレーニング) = 0.17
: 920. 正答率(テスト) = 0.101
Generation: 930. 正答率(トレーニング) = 0.14
: 930. 正答率(テスト) = 0.101
Generation: 940. 正答率(トレーニング) = 0.13
: 940. 正答率(テスト) = 0.0958
Generation: 950. 正答率(トレーニング) = 0.17
: 950. 正答率(テスト) = 0.0892
Generation: 960. 正答率(トレーニング) = 0.11
: 960. 正答率(テスト) = 0.101
Generation: 970. 正答率(トレーニング) = 0.12
: 970. 正答率(テスト) = 0.0982
Generation: 980. 正答率(トレーニング) = 0.17
: 980. 正答率(テスト) = 0.1028
Generation: 990. 正答率(トレーニング) = 0.14
: 990. 正答率(テスト) = 0.098
Generation: 1000. 正答率(トレーニング) = 0.11
: 1000. 正答率(テスト) = 0.1032



[try] 活性化関数と重みの初期化方法を変えてみよう

AdaGrad

```
In [38]: # AdaGradを作ってみよう
# データの読み込み
(x_train, d_train), (x_test, d_test) = load_mnist(normalize=True, one_hot_label=True)

print("データ読み込み完了")

# batch_normalizationの設定 =====
#use_batchnorm = True
use_batchnorm = False
# =====

network = MultiLayerNet(input_size=784, hidden_size_list=[40, 20], output_size=10, activation='relu', weight_init_std='he',
                        use_batchnorm=use_batchnorm)

iters_num = 1000
# iters_num = 500 # 処理を短縮

train_size = x_train.shape[0]
batch_size = 100
learning_rate = 0.01

# AdaGradでは不必要
# =====

#momentum = 0.9

# =====

train_loss_list = []
accuracies_train = []
accuracies_test = []

plot_interval=10

for i in range(iters_num):
    batch_mask = np.random.choice(train_size, batch_size)
    x_batch = x_train[batch_mask]
    d_batch = d_train[batch_mask]

    # 勾配
    grad = network.gradient(x_batch, d_batch)
```

```

if i == 0:
    h = {}
for key in ('W1', 'W2', 'W3', 'b1', 'b2', 'b3'):

    # 変更しよう
    # =====
    if i == 0:
        h[key] = np.zeros_like(network.params[key])+1e-4
        h[key] = h[key] + grad[key] * grad[key]
        network.params[key] -= learning_rate * grad[key] / (np.sqrt(h[key])+ 1e-4 )

    # =====

    loss = network.loss(x_batch, d_batch)
    train_loss_list.append(loss)

if (i + 1) % plot_interval == 0:
    accr_test = network.accuracy(x_test, d_test)
    accuracies_test.append(accr_test)
    accr_train = network.accuracy(x_batch, d_batch)
    accuracies_train.append(accr_train)

    print('Generation: ' + str(i+1) + '. 正答率(トレーニング) = ' + str(accr_train))
    print('          : ' + str(i+1) + '. 正答率(テスト) = ' + str(accr_test))

lists = range(0, iters_num, plot_interval)
plt.plot(lists, accuracies_train, label="training set")
plt.plot(lists, accuracies_test, label="test set")
plt.legend(loc="lower right")
plt.title("accuracy")
plt.xlabel("count")
plt.ylabel("accuracy")
plt.ylim(0, 1.0)
# グラフの表示
plt.show()

```

データ読み込み完了

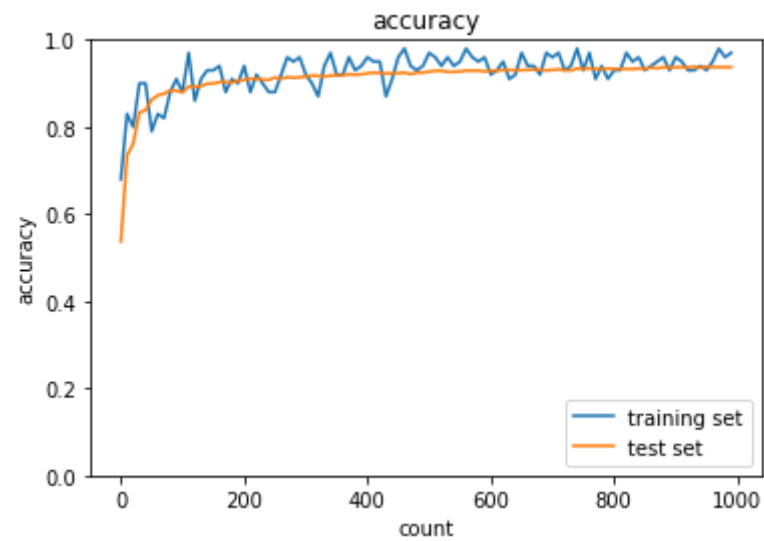
Generation: 10. 正答率(トレーニング) = 0.68
: 10. 正答率(テスト) = 0.5373
Generation: 20. 正答率(トレーニング) = 0.83
: 20. 正答率(テスト) = 0.7345
Generation: 30. 正答率(トレーニング) = 0.8
: 30. 正答率(テスト) = 0.7618
Generation: 40. 正答率(トレーニング) = 0.9
: 40. 正答率(テスト) = 0.833
Generation: 50. 正答率(トレーニング) = 0.9
: 50. 正答率(テスト) = 0.8388
Generation: 60. 正答率(トレーニング) = 0.79
: 60. 正答率(テスト) = 0.8623
Generation: 70. 正答率(トレーニング) = 0.83
: 70. 正答率(テスト) = 0.8729
Generation: 80. 正答率(トレーニング) = 0.82
: 80. 正答率(テスト) = 0.8762
Generation: 90. 正答率(トレーニング) = 0.88
: 90. 正答率(テスト) = 0.8839
Generation: 100. 正答率(トレーニング) = 0.91
: 100. 正答率(テスト) = 0.8831
Generation: 110. 正答率(トレーニング) = 0.88
: 110. 正答率(テスト) = 0.8791
Generation: 120. 正答率(トレーニング) = 0.97
: 120. 正答率(テスト) = 0.8928
Generation: 130. 正答率(トレーニング) = 0.86
: 130. 正答率(テスト) = 0.8923
Generation: 140. 正答率(トレーニング) = 0.91
: 140. 正答率(テスト) = 0.8937
Generation: 150. 正答率(トレーニング) = 0.93
: 150. 正答率(テスト) = 0.8993
Generation: 160. 正答率(トレーニング) = 0.93
: 160. 正答率(テスト) = 0.8997
Generation: 170. 正答率(トレーニング) = 0.94
: 170. 正答率(テスト) = 0.9027
Generation: 180. 正答率(トレーニング) = 0.88
: 180. 正答率(テスト) = 0.9039
Generation: 190. 正答率(トレーニング) = 0.91
: 190. 正答率(テスト) = 0.9015
Generation: 200. 正答率(トレーニング) = 0.9
: 200. 正答率(テスト) = 0.9064

Generation: 210. 正答率(トレーニング) = 0.94
: 210. 正答率(テスト) = 0.9076
Generation: 220. 正答率(トレーニング) = 0.88
: 220. 正答率(テスト) = 0.9107
Generation: 230. 正答率(トレーニング) = 0.92
: 230. 正答率(テスト) = 0.9109
Generation: 240. 正答率(トレーニング) = 0.9
: 240. 正答率(テスト) = 0.9086
Generation: 250. 正答率(トレーニング) = 0.88
: 250. 正答率(テスト) = 0.9079
Generation: 260. 正答率(トレーニング) = 0.88
: 260. 正答率(テスト) = 0.9135
Generation: 270. 正答率(トレーニング) = 0.92
: 270. 正答率(テスト) = 0.9105
Generation: 280. 正答率(トレーニング) = 0.96
: 280. 正答率(テスト) = 0.9142
Generation: 290. 正答率(トレーニング) = 0.95
: 290. 正答率(テスト) = 0.9128
Generation: 300. 正答率(トレーニング) = 0.96
: 300. 正答率(テスト) = 0.9139
Generation: 310. 正答率(トレーニング) = 0.92
: 310. 正答率(テスト) = 0.9158
Generation: 320. 正答率(トレーニング) = 0.9
: 320. 正答率(テスト) = 0.9178
Generation: 330. 正答率(トレーニング) = 0.87
: 330. 正答率(テスト) = 0.9172
Generation: 340. 正答率(トレーニング) = 0.94
: 340. 正答率(テスト) = 0.9158
Generation: 350. 正答率(トレーニング) = 0.97
: 350. 正答率(テスト) = 0.9178
Generation: 360. 正答率(トレーニング) = 0.92
: 360. 正答率(テスト) = 0.9186
Generation: 370. 正答率(トレーニング) = 0.92
: 370. 正答率(テスト) = 0.9185
Generation: 380. 正答率(トレーニング) = 0.96
: 380. 正答率(テスト) = 0.921
Generation: 390. 正答率(トレーニング) = 0.93
: 390. 正答率(テスト) = 0.9198
Generation: 400. 正答率(トレーニング) = 0.94
: 400. 正答率(テスト) = 0.9207
Generation: 410. 正答率(トレーニング) = 0.96
: 410. 正答率(テスト) = 0.923

Generation: 420. 正答率(トレーニング) = 0.95
: 420. 正答率(テスト) = 0.9246
Generation: 430. 正答率(トレーニング) = 0.95
: 430. 正答率(テスト) = 0.9237
Generation: 440. 正答率(トレーニング) = 0.87
: 440. 正答率(テスト) = 0.9229
Generation: 450. 正答率(トレーニング) = 0.91
: 450. 正答率(テスト) = 0.9226
Generation: 460. 正答率(トレーニング) = 0.96
: 460. 正答率(テスト) = 0.9235
Generation: 470. 正答率(トレーニング) = 0.98
: 470. 正答率(テスト) = 0.9246
Generation: 480. 正答率(トレーニング) = 0.94
: 480. 正答率(テスト) = 0.9212
Generation: 490. 正答率(トレーニング) = 0.93
: 490. 正答率(テスト) = 0.9244
Generation: 500. 正答率(トレーニング) = 0.94
: 500. 正答率(テスト) = 0.9249
Generation: 510. 正答率(トレーニング) = 0.97
: 510. 正答率(テスト) = 0.9269
Generation: 520. 正答率(トレーニング) = 0.96
: 520. 正答率(テスト) = 0.929
Generation: 530. 正答率(トレーニング) = 0.94
: 530. 正答率(テスト) = 0.9291
Generation: 540. 正答率(トレーニング) = 0.96
: 540. 正答率(テスト) = 0.9264
Generation: 550. 正答率(トレーニング) = 0.94
: 550. 正答率(テスト) = 0.9262
Generation: 560. 正答率(トレーニング) = 0.95
: 560. 正答率(テスト) = 0.9275
Generation: 570. 正答率(トレーニング) = 0.98
: 570. 正答率(テスト) = 0.929
Generation: 580. 正答率(トレーニング) = 0.96
: 580. 正答率(テスト) = 0.9293
Generation: 590. 正答率(トレーニング) = 0.95
: 590. 正答率(テスト) = 0.9292
Generation: 600. 正答率(トレーニング) = 0.96
: 600. 正答率(テスト) = 0.9273
Generation: 610. 正答率(トレーニング) = 0.92
: 610. 正答率(テスト) = 0.9291
Generation: 620. 正答率(トレーニング) = 0.93
: 620. 正答率(テスト) = 0.9289

Generation: 630. 正答率(トレーニング) = 0.95
: 630. 正答率(テスト) = 0.9306
Generation: 640. 正答率(トレーニング) = 0.91
: 640. 正答率(テスト) = 0.9307
Generation: 650. 正答率(トレーニング) = 0.92
: 650. 正答率(テスト) = 0.9302
Generation: 660. 正答率(トレーニング) = 0.97
: 660. 正答率(テスト) = 0.9303
Generation: 670. 正答率(トレーニング) = 0.94
: 670. 正答率(テスト) = 0.9311
Generation: 680. 正答率(トレーニング) = 0.94
: 680. 正答率(テスト) = 0.9313
Generation: 690. 正答率(トレーニング) = 0.92
: 690. 正答率(テスト) = 0.9313
Generation: 700. 正答率(トレーニング) = 0.97
: 700. 正答率(テスト) = 0.9297
Generation: 710. 正答率(トレーニング) = 0.96
: 710. 正答率(テスト) = 0.9308
Generation: 720. 正答率(トレーニング) = 0.97
: 720. 正答率(テスト) = 0.9322
Generation: 730. 正答率(トレーニング) = 0.93
: 730. 正答率(テスト) = 0.9301
Generation: 740. 正答率(トレーニング) = 0.94
: 740. 正答率(テスト) = 0.9296
Generation: 750. 正答率(トレーニング) = 0.98
: 750. 正答率(テスト) = 0.9343
Generation: 760. 正答率(トレーニング) = 0.93
: 760. 正答率(テスト) = 0.9335
Generation: 770. 正答率(トレーニング) = 0.97
: 770. 正答率(テスト) = 0.9342
Generation: 780. 正答率(トレーニング) = 0.91
: 780. 正答率(テスト) = 0.9332
Generation: 790. 正答率(トレーニング) = 0.94
: 790. 正答率(テスト) = 0.9331
Generation: 800. 正答率(トレーニング) = 0.91
: 800. 正答率(テスト) = 0.9341
Generation: 810. 正答率(トレーニング) = 0.93
: 810. 正答率(テスト) = 0.9333
Generation: 820. 正答率(トレーニング) = 0.93
: 820. 正答率(テスト) = 0.9329
Generation: 830. 正答率(トレーニング) = 0.97
: 830. 正答率(テスト) = 0.9331

Generation: 840. 正答率(トレーニング) = 0.95
: 840. 正答率(テスト) = 0.9328
Generation: 850. 正答率(トレーニング) = 0.96
: 850. 正答率(テスト) = 0.9344
Generation: 860. 正答率(トレーニング) = 0.93
: 860. 正答率(テスト) = 0.9332
Generation: 870. 正答率(トレーニング) = 0.94
: 870. 正答率(テスト) = 0.9364
Generation: 880. 正答率(トレーニング) = 0.95
: 880. 正答率(テスト) = 0.9348
Generation: 890. 正答率(トレーニング) = 0.96
: 890. 正答率(テスト) = 0.9369
Generation: 900. 正答率(トレーニング) = 0.93
: 900. 正答率(テスト) = 0.9363
Generation: 910. 正答率(トレーニング) = 0.96
: 910. 正答率(テスト) = 0.9366
Generation: 920. 正答率(トレーニング) = 0.95
: 920. 正答率(テスト) = 0.9375
Generation: 930. 正答率(トレーニング) = 0.93
: 930. 正答率(テスト) = 0.9363
Generation: 940. 正答率(トレーニング) = 0.93
: 940. 正答率(テスト) = 0.9384
Generation: 950. 正答率(トレーニング) = 0.94
: 950. 正答率(テスト) = 0.9357
Generation: 960. 正答率(トレーニング) = 0.93
: 960. 正答率(テスト) = 0.9382
Generation: 970. 正答率(トレーニング) = 0.95
: 970. 正答率(テスト) = 0.9375
Generation: 980. 正答率(トレーニング) = 0.98
: 980. 正答率(テスト) = 0.9373
Generation: 990. 正答率(トレーニング) = 0.96
: 990. 正答率(テスト) = 0.9372
Generation: 1000. 正答率(トレーニング) = 0.97
: 1000. 正答率(テスト) = 0.9373



[try] バッチ正規化をしてみよう & ReLU, He

AdaGrad

```
In [50]: # AdaGradを作ってみよう
# データの読み込み
(x_train, d_train), (x_test, d_test) = load_mnist(normalize=True, one_hot_label=True)

print("データ読み込み完了")

# batch_normalizationの設定 =====
use_batchnorm = True
#use_batchnorm = False
# =====

network = MultiLayerNet(input_size=784, hidden_size_list=[40, 20], output_size=10, activation='relu', weight_init_std='he',
                        use_batchnorm=use_batchnorm)

iters_num = 1000
# iters_num = 500 # 処理を短縮

train_size = x_train.shape[0]
batch_size = 100
learning_rate = 0.01

# AdaGradでは不必要
# =====

#momentum = 0.9

# =====

train_loss_list = []
accuracies_train = []
accuracies_test = []

plot_interval=10

for i in range(iters_num):
    batch_mask = np.random.choice(train_size, batch_size)
    x_batch = x_train[batch_mask]
    d_batch = d_train[batch_mask]

    # 勾配
    grad = network.gradient(x_batch, d_batch)
```

```

if i == 0:
    h = {}
for key in ('W1', 'W2', 'W3', 'b1', 'b2', 'b3'):

    # 変更しよう
    # =====
    if i == 0:
        h[key] = np.zeros_like(network.params[key])+1e-4
    h[key] = h[key] + grad[key] * grad[key]
    network.params[key] -= learning_rate * grad[key] / (np.sqrt(h[key])+ 1e-4 )

    # =====

    loss = network.loss(x_batch, d_batch)
    train_loss_list.append(loss)

if (i + 1) % plot_interval == 0:
    accr_test = network.accuracy(x_test, d_test)
    accuracies_test.append(accr_test)
    accr_train = network.accuracy(x_batch, d_batch)
    accuracies_train.append(accr_train)

    print('Generation: ' + str(i+1) + '. 正答率(トレーニング) = ' + str(accr_train))
    print('          : ' + str(i+1) + '. 正答率(テスト) = ' + str(accr_test))

lists = range(0, iters_num, plot_interval)
plt.plot(lists, accuracies_train, label="training set")
plt.plot(lists, accuracies_test, label="test set")
plt.legend(loc="lower right")
plt.title("accuracy")
plt.xlabel("count")
plt.ylabel("accuracy")
plt.ylim(0, 1.0)
# グラフの表示
plt.show()

```

データ読み込み完了

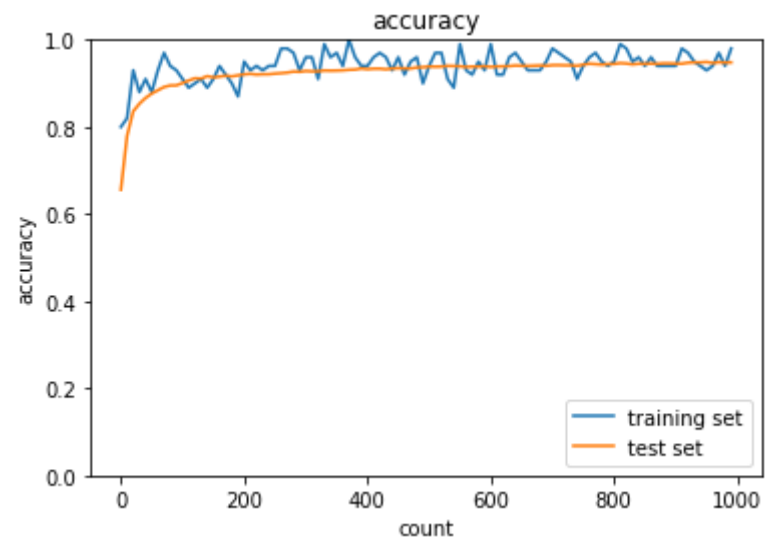
Generation: 10. 正答率(トレーニング) = 0.8
: 10. 正答率(テスト) = 0.656
Generation: 20. 正答率(トレーニング) = 0.82
: 20. 正答率(テスト) = 0.7793
Generation: 30. 正答率(トレーニング) = 0.93
: 30. 正答率(テスト) = 0.8359
Generation: 40. 正答率(トレーニング) = 0.88
: 40. 正答率(テスト) = 0.8532
Generation: 50. 正答率(トレーニング) = 0.91
: 50. 正答率(テスト) = 0.8665
Generation: 60. 正答率(トレーニング) = 0.88
: 60. 正答率(テスト) = 0.8766
Generation: 70. 正答率(トレーニング) = 0.93
: 70. 正答率(テスト) = 0.8842
Generation: 80. 正答率(トレーニング) = 0.97
: 80. 正答率(テスト) = 0.8916
Generation: 90. 正答率(トレーニング) = 0.94
: 90. 正答率(テスト) = 0.8952
Generation: 100. 正答率(トレーニング) = 0.93
: 100. 正答率(テスト) = 0.8954
Generation: 110. 正答率(トレーニング) = 0.91
: 110. 正答率(テスト) = 0.9008
Generation: 120. 正答率(トレーニング) = 0.89
: 120. 正答率(テスト) = 0.9064
Generation: 130. 正答率(トレーニング) = 0.9
: 130. 正答率(テスト) = 0.9117
Generation: 140. 正答率(トレーニング) = 0.91
: 140. 正答率(テスト) = 0.9109
Generation: 150. 正答率(トレーニング) = 0.89
: 150. 正答率(テスト) = 0.9165
Generation: 160. 正答率(トレーニング) = 0.91
: 160. 正答率(テスト) = 0.9137
Generation: 170. 正答率(トレーニング) = 0.94
: 170. 正答率(テスト) = 0.9155
Generation: 180. 正答率(トレーニング) = 0.92
: 180. 正答率(テスト) = 0.9176
Generation: 190. 正答率(トレーニング) = 0.9
: 190. 正答率(テスト) = 0.916
Generation: 200. 正答率(トレーニング) = 0.87
: 200. 正答率(テスト) = 0.9185

Generation: 210. 正答率(トレーニング) = 0.95
: 210. 正答率(テスト) = 0.9207
Generation: 220. 正答率(トレーニング) = 0.93
: 220. 正答率(テスト) = 0.9219
Generation: 230. 正答率(トレーニング) = 0.94
: 230. 正答率(テスト) = 0.9202
Generation: 240. 正答率(トレーニング) = 0.93
: 240. 正答率(テスト) = 0.9209
Generation: 250. 正答率(トレーニング) = 0.94
: 250. 正答率(テスト) = 0.921
Generation: 260. 正答率(トレーニング) = 0.94
: 260. 正答率(テスト) = 0.9224
Generation: 270. 正答率(トレーニング) = 0.98
: 270. 正答率(テスト) = 0.9234
Generation: 280. 正答率(トレーニング) = 0.98
: 280. 正答率(テスト) = 0.9247
Generation: 290. 正答率(トレーニング) = 0.97
: 290. 正答率(テスト) = 0.9271
Generation: 300. 正答率(トレーニング) = 0.93
: 300. 正答率(テスト) = 0.9265
Generation: 310. 正答率(トレーニング) = 0.96
: 310. 正答率(テスト) = 0.9284
Generation: 320. 正答率(トレーニング) = 0.96
: 320. 正答率(テスト) = 0.9276
Generation: 330. 正答率(トレーニング) = 0.91
: 330. 正答率(テスト) = 0.9277
Generation: 340. 正答率(トレーニング) = 0.99
: 340. 正答率(テスト) = 0.9295
Generation: 350. 正答率(トレーニング) = 0.96
: 350. 正答率(テスト) = 0.9291
Generation: 360. 正答率(トレーニング) = 0.97
: 360. 正答率(テスト) = 0.9294
Generation: 370. 正答率(トレーニング) = 0.94
: 370. 正答率(テスト) = 0.9299
Generation: 380. 正答率(トレーニング) = 1.0
: 380. 正答率(テスト) = 0.9312
Generation: 390. 正答率(トレーニング) = 0.96
: 390. 正答率(テスト) = 0.9317
Generation: 400. 正答率(トレーニング) = 0.94
: 400. 正答率(テスト) = 0.9341
Generation: 410. 正答率(トレーニング) = 0.94
: 410. 正答率(テスト) = 0.9326

Generation: 420. 正答率(トレーニング) = 0.96
: 420. 正答率(テスト) = 0.9336
Generation: 430. 正答率(トレーニング) = 0.97
: 430. 正答率(テスト) = 0.9337
Generation: 440. 正答率(トレーニング) = 0.96
: 440. 正答率(テスト) = 0.9323
Generation: 450. 正答率(トレーニング) = 0.93
: 450. 正答率(テスト) = 0.9341
Generation: 460. 正答率(トレーニング) = 0.96
: 460. 正答率(テスト) = 0.9348
Generation: 470. 正答率(トレーニング) = 0.92
: 470. 正答率(テスト) = 0.9343
Generation: 480. 正答率(トレーニング) = 0.95
: 480. 正答率(テスト) = 0.9341
Generation: 490. 正答率(トレーニング) = 0.96
: 490. 正答率(テスト) = 0.9365
Generation: 500. 正答率(トレーニング) = 0.9
: 500. 正答率(テスト) = 0.9372
Generation: 510. 正答率(トレーニング) = 0.94
: 510. 正答率(テスト) = 0.9386
Generation: 520. 正答率(トレーニング) = 0.97
: 520. 正答率(テスト) = 0.9382
Generation: 530. 正答率(トレーニング) = 0.97
: 530. 正答率(テスト) = 0.9386
Generation: 540. 正答率(トレーニング) = 0.91
: 540. 正答率(テスト) = 0.9401
Generation: 550. 正答率(トレーニング) = 0.89
: 550. 正答率(テスト) = 0.9398
Generation: 560. 正答率(トレーニング) = 0.99
: 560. 正答率(テスト) = 0.9392
Generation: 570. 正答率(トレーニング) = 0.93
: 570. 正答率(テスト) = 0.9377
Generation: 580. 正答率(トレーニング) = 0.92
: 580. 正答率(テスト) = 0.9384
Generation: 590. 正答率(トレーニング) = 0.95
: 590. 正答率(テスト) = 0.9386
Generation: 600. 正答率(トレーニング) = 0.93
: 600. 正答率(テスト) = 0.9392
Generation: 610. 正答率(トレーニング) = 0.99
: 610. 正答率(テスト) = 0.9378
Generation: 620. 正答率(トレーニング) = 0.92
: 620. 正答率(テスト) = 0.9388

Generation: 630. 正答率(トレーニング) = 0.92
: 630. 正答率(テスト) = 0.9391
Generation: 640. 正答率(トレーニング) = 0.96
: 640. 正答率(テスト) = 0.9396
Generation: 650. 正答率(トレーニング) = 0.97
: 650. 正答率(テスト) = 0.941
Generation: 660. 正答率(トレーニング) = 0.95
: 660. 正答率(テスト) = 0.9404
Generation: 670. 正答率(トレーニング) = 0.93
: 670. 正答率(テスト) = 0.94
Generation: 680. 正答率(トレーニング) = 0.93
: 680. 正答率(テスト) = 0.9405
Generation: 690. 正答率(トレーニング) = 0.93
: 690. 正答率(テスト) = 0.9411
Generation: 700. 正答率(トレーニング) = 0.95
: 700. 正答率(テスト) = 0.9403
Generation: 710. 正答率(トレーニング) = 0.98
: 710. 正答率(テスト) = 0.9418
Generation: 720. 正答率(トレーニング) = 0.97
: 720. 正答率(テスト) = 0.9419
Generation: 730. 正答率(トレーニング) = 0.96
: 730. 正答率(テスト) = 0.9415
Generation: 740. 正答率(トレーニング) = 0.95
: 740. 正答率(テスト) = 0.9415
Generation: 750. 正答率(トレーニング) = 0.91
: 750. 正答率(テスト) = 0.9403
Generation: 760. 正答率(トレーニング) = 0.94
: 760. 正答率(テスト) = 0.9429
Generation: 770. 正答率(トレーニング) = 0.96
: 770. 正答率(テスト) = 0.9452
Generation: 780. 正答率(トレーニング) = 0.97
: 780. 正答率(テスト) = 0.9442
Generation: 790. 正答率(トレーニング) = 0.95
: 790. 正答率(テスト) = 0.9431
Generation: 800. 正答率(トレーニング) = 0.94
: 800. 正答率(テスト) = 0.9443
Generation: 810. 正答率(トレーニング) = 0.95
: 810. 正答率(テスト) = 0.9446
Generation: 820. 正答率(トレーニング) = 0.99
: 820. 正答率(テスト) = 0.9465
Generation: 830. 正答率(トレーニング) = 0.98
: 830. 正答率(テスト) = 0.9456

Generation: 840. 正答率(トレーニング) = 0.95
: 840. 正答率(テスト) = 0.9436
Generation: 850. 正答率(トレーニング) = 0.96
: 850. 正答率(テスト) = 0.9451
Generation: 860. 正答率(トレーニング) = 0.94
: 860. 正答率(テスト) = 0.9448
Generation: 870. 正答率(トレーニング) = 0.96
: 870. 正答率(テスト) = 0.9454
Generation: 880. 正答率(トレーニング) = 0.94
: 880. 正答率(テスト) = 0.9448
Generation: 890. 正答率(トレーニング) = 0.94
: 890. 正答率(テスト) = 0.9459
Generation: 900. 正答率(トレーニング) = 0.94
: 900. 正答率(テスト) = 0.9458
Generation: 910. 正答率(トレーニング) = 0.94
: 910. 正答率(テスト) = 0.9452
Generation: 920. 正答率(トレーニング) = 0.98
: 920. 正答率(テスト) = 0.9452
Generation: 930. 正答率(トレーニング) = 0.97
: 930. 正答率(テスト) = 0.9474
Generation: 940. 正答率(トレーニング) = 0.95
: 940. 正答率(テスト) = 0.9481
Generation: 950. 正答率(トレーニング) = 0.94
: 950. 正答率(テスト) = 0.948
Generation: 960. 正答率(トレーニング) = 0.93
: 960. 正答率(テスト) = 0.9491
Generation: 970. 正答率(トレーニング) = 0.94
: 970. 正答率(テスト) = 0.9467
Generation: 980. 正答率(トレーニング) = 0.97
: 980. 正答率(テスト) = 0.9486
Generation: 990. 正答率(トレーニング) = 0.94
: 990. 正答率(テスト) = 0.9476
Generation: 1000. 正答率(トレーニング) = 0.98
: 1000. 正答率(テスト) = 0.9481



RSMprop

```
In [24]: # データの読み込み
(x_train, d_train), (x_test, d_test) = load_mnist(normalize=True, one_hot_label=True)

print("データ読み込み完了")

# batch_normalizationの設定 =====
#use_batchnorm = True
use_batchnorm = False
# =====

network = MultiLayerNet(input_size=784, hidden_size_list=[40, 20], output_size=10, activation='sigmoid', weight_init_std=0.01,
                        use_batchnorm=use_batchnorm)

iters_num = 1000
train_size = x_train.shape[0]
batch_size = 100
learning_rate = 0.01
decay_rate = 0.99

train_loss_list = []
accuracies_train = []
accuracies_test = []

plot_interval=10

for i in range(iters_num):
    batch_mask = np.random.choice(train_size, batch_size)
    x_batch = x_train[batch_mask]
    d_batch = d_train[batch_mask]

    # 勾配
    grad = network.gradient(x_batch, d_batch)
    if i == 0:
        h = {}
    for key in ('W1', 'W2', 'W3', 'b1', 'b2', 'b3'):
        if i == 0:
            h[key] = np.zeros_like(network.params[key])
        h[key] *= decay_rate
        h[key] += (1 - decay_rate) * np.square(grad[key])
        network.params[key] -= learning_rate * grad[key] / (np.sqrt(h[key]) + 1e-7)
```

```
    loss = network.loss(x_batch, d_batch)
    train_loss_list.append(loss)

    if (i + 1) % plot_interval == 0:
        accr_test = network.accuracy(x_test, d_test)
        accuracies_test.append(accr_test)
        accr_train = network.accuracy(x_batch, d_batch)
        accuracies_train.append(accr_train)

        print('Generation: ' + str(i+1) + '. 正答率(トレーニング) = ' + str(accr_train))
        print('                : ' + str(i+1) + '. 正答率(テスト) = ' + str(accr_test))

lists = range(0, iters_num, plot_interval)
plt.plot(lists, accuracies_train, label="training set")
plt.plot(lists, accuracies_test, label="test set")
plt.legend(loc="lower right")
plt.title("accuracy")
plt.xlabel("count")
plt.ylabel("accuracy")
plt.ylim(0, 1.0)
# グラフの表示
plt.show()
```

データ読み込み完了

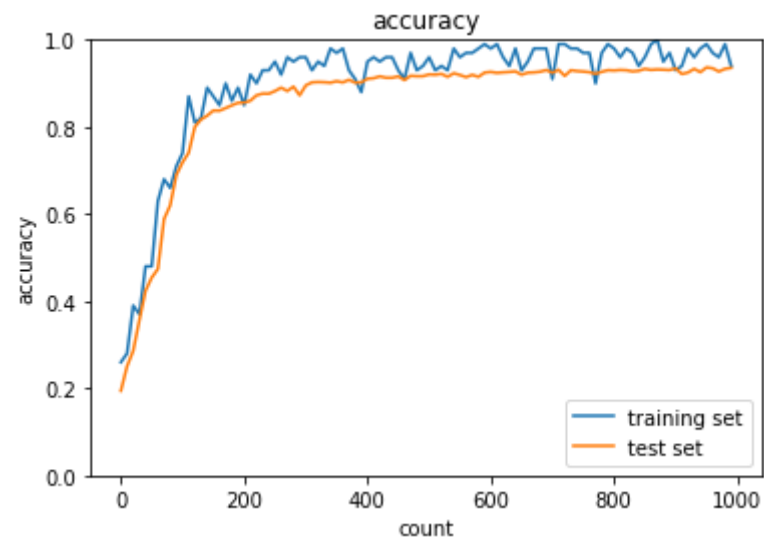
Generation: 10. 正答率(トレーニング) = 0.26
: 10. 正答率(テスト) = 0.1944
Generation: 20. 正答率(トレーニング) = 0.28
: 20. 正答率(テスト) = 0.2506
Generation: 30. 正答率(トレーニング) = 0.39
: 30. 正答率(テスト) = 0.286
Generation: 40. 正答率(トレーニング) = 0.37
: 40. 正答率(テスト) = 0.3563
Generation: 50. 正答率(トレーニング) = 0.48
: 50. 正答率(テスト) = 0.4241
Generation: 60. 正答率(トレーニング) = 0.48
: 60. 正答率(テスト) = 0.4541
Generation: 70. 正答率(トレーニング) = 0.63
: 70. 正答率(テスト) = 0.4738
Generation: 80. 正答率(トレーニング) = 0.68
: 80. 正答率(テスト) = 0.5885
Generation: 90. 正答率(トレーニング) = 0.66
: 90. 正答率(テスト) = 0.6201
Generation: 100. 正答率(トレーニング) = 0.71
: 100. 正答率(テスト) = 0.6899
Generation: 110. 正答率(トレーニング) = 0.74
: 110. 正答率(テスト) = 0.7186
Generation: 120. 正答率(トレーニング) = 0.87
: 120. 正答率(テスト) = 0.7408
Generation: 130. 正答率(トレーニング) = 0.81
: 130. 正答率(テスト) = 0.8002
Generation: 140. 正答率(トレーニング) = 0.82
: 140. 正答率(テスト) = 0.8171
Generation: 150. 正答率(トレーニング) = 0.89
: 150. 正答率(テスト) = 0.8259
Generation: 160. 正答率(トレーニング) = 0.87
: 160. 正答率(テスト) = 0.8376
Generation: 170. 正答率(トレーニング) = 0.85
: 170. 正答率(テスト) = 0.8373
Generation: 180. 正答率(トレーニング) = 0.9
: 180. 正答率(テスト) = 0.8433
Generation: 190. 正答率(トレーニング) = 0.86
: 190. 正答率(テスト) = 0.8496
Generation: 200. 正答率(トレーニング) = 0.89
: 200. 正答率(テスト) = 0.8553

Generation: 210. 正答率(トレーニング) = 0.85
: 210. 正答率(テスト) = 0.8559
Generation: 220. 正答率(トレーニング) = 0.92
: 220. 正答率(テスト) = 0.8601
Generation: 230. 正答率(トレーニング) = 0.9
: 230. 正答率(テスト) = 0.8725
Generation: 240. 正答率(トレーニング) = 0.93
: 240. 正答率(テスト) = 0.8765
Generation: 250. 正答率(トレーニング) = 0.93
: 250. 正答率(テスト) = 0.8763
Generation: 260. 正答率(トレーニング) = 0.95
: 260. 正答率(テスト) = 0.8824
Generation: 270. 正答率(トレーニング) = 0.92
: 270. 正答率(テスト) = 0.8901
Generation: 280. 正答率(トレーニング) = 0.96
: 280. 正答率(テスト) = 0.8825
Generation: 290. 正答率(トレーニング) = 0.95
: 290. 正答率(テスト) = 0.8924
Generation: 300. 正答率(トレーニング) = 0.96
: 300. 正答率(テスト) = 0.8726
Generation: 310. 正答率(トレーニング) = 0.96
: 310. 正答率(テスト) = 0.8954
Generation: 320. 正答率(トレーニング) = 0.93
: 320. 正答率(テスト) = 0.9023
Generation: 330. 正答率(トレーニング) = 0.95
: 330. 正答率(テスト) = 0.9031
Generation: 340. 正答率(トレーニング) = 0.94
: 340. 正答率(テスト) = 0.9026
Generation: 350. 正答率(トレーニング) = 0.98
: 350. 正答率(テスト) = 0.9015
Generation: 360. 正答率(トレーニング) = 0.97
: 360. 正答率(テスト) = 0.905
Generation: 370. 正答率(トレーニング) = 0.98
: 370. 正答率(テスト) = 0.9022
Generation: 380. 正答率(トレーニング) = 0.93
: 380. 正答率(テスト) = 0.9082
Generation: 390. 正答率(トレーニング) = 0.91
: 390. 正答率(テスト) = 0.9016
Generation: 400. 正答率(トレーニング) = 0.88
: 400. 正答率(テスト) = 0.9022
Generation: 410. 正答率(トレーニング) = 0.95
: 410. 正答率(テスト) = 0.9105

Generation: 420. 正答率(トレーニング) = 0.96
: 420. 正答率(テスト) = 0.9118
Generation: 430. 正答率(トレーニング) = 0.95
: 430. 正答率(テスト) = 0.916
Generation: 440. 正答率(トレーニング) = 0.96
: 440. 正答率(テスト) = 0.9129
Generation: 450. 正答率(トレーニング) = 0.96
: 450. 正答率(テスト) = 0.9129
Generation: 460. 正答率(トレーニング) = 0.93
: 460. 正答率(テスト) = 0.916
Generation: 470. 正答率(トレーニング) = 0.91
: 470. 正答率(テスト) = 0.9076
Generation: 480. 正答率(トレーニング) = 0.97
: 480. 正答率(テスト) = 0.9174
Generation: 490. 正答率(トレーニング) = 0.93
: 490. 正答率(テスト) = 0.9164
Generation: 500. 正答率(トレーニング) = 0.94
: 500. 正答率(テスト) = 0.9165
Generation: 510. 正答率(トレーニング) = 0.96
: 510. 正答率(テスト) = 0.9205
Generation: 520. 正答率(トレーニング) = 0.93
: 520. 正答率(テスト) = 0.9198
Generation: 530. 正答率(トレーニング) = 0.94
: 530. 正答率(テスト) = 0.9218
Generation: 540. 正答率(トレーニング) = 0.93
: 540. 正答率(テスト) = 0.915
Generation: 550. 正答率(トレーニング) = 0.98
: 550. 正答率(テスト) = 0.9232
Generation: 560. 正答率(トレーニング) = 0.96
: 560. 正答率(テスト) = 0.9192
Generation: 570. 正答率(トレーニング) = 0.97
: 570. 正答率(テスト) = 0.9143
Generation: 580. 正答率(トレーニング) = 0.97
: 580. 正答率(テスト) = 0.9195
Generation: 590. 正答率(トレーニング) = 0.98
: 590. 正答率(テスト) = 0.9149
Generation: 600. 正答率(トレーニング) = 0.99
: 600. 正答率(テスト) = 0.9247
Generation: 610. 正答率(トレーニング) = 0.98
: 610. 正答率(テスト) = 0.9263
Generation: 620. 正答率(トレーニング) = 0.99
: 620. 正答率(テスト) = 0.9241

Generation: 630. 正答率(トレーニング) = 0.96
: 630. 正答率(テスト) = 0.9253
Generation: 640. 正答率(トレーニング) = 0.94
: 640. 正答率(テスト) = 0.9262
Generation: 650. 正答率(トレーニング) = 0.98
: 650. 正答率(テスト) = 0.9275
Generation: 660. 正答率(トレーニング) = 0.93
: 660. 正答率(テスト) = 0.9198
Generation: 670. 正答率(トレーニング) = 0.95
: 670. 正答率(テスト) = 0.9245
Generation: 680. 正答率(トレーニング) = 0.98
: 680. 正答率(テスト) = 0.9252
Generation: 690. 正答率(トレーニング) = 0.98
: 690. 正答率(テスト) = 0.9267
Generation: 700. 正答率(トレーニング) = 0.98
: 700. 正答率(テスト) = 0.9305
Generation: 710. 正答率(トレーニング) = 0.91
: 710. 正答率(テスト) = 0.9244
Generation: 720. 正答率(トレーニング) = 0.99
: 720. 正答率(テスト) = 0.931
Generation: 730. 正答率(トレーニング) = 0.99
: 730. 正答率(テスト) = 0.9172
Generation: 740. 正答率(トレーニング) = 0.98
: 740. 正答率(テスト) = 0.9301
Generation: 750. 正答率(トレーニング) = 0.98
: 750. 正答率(テスト) = 0.9282
Generation: 760. 正答率(トレーニング) = 0.97
: 760. 正答率(テスト) = 0.9272
Generation: 770. 正答率(トレーニング) = 0.97
: 770. 正答率(テスト) = 0.9256
Generation: 780. 正答率(トレーニング) = 0.9
: 780. 正答率(テスト) = 0.9219
Generation: 790. 正答率(トレーニング) = 0.97
: 790. 正答率(テスト) = 0.9265
Generation: 800. 正答率(トレーニング) = 0.99
: 800. 正答率(テスト) = 0.9302
Generation: 810. 正答率(トレーニング) = 0.98
: 810. 正答率(テスト) = 0.9295
Generation: 820. 正答率(トレーニング) = 0.96
: 820. 正答率(テスト) = 0.9307
Generation: 830. 正答率(トレーニング) = 0.98
: 830. 正答率(テスト) = 0.9301

Generation: 840. 正答率(トレーニング) = 0.97
: 840. 正答率(テスト) = 0.9275
Generation: 850. 正答率(トレーニング) = 0.94
: 850. 正答率(テスト) = 0.9288
Generation: 860. 正答率(トレーニング) = 0.96
: 860. 正答率(テスト) = 0.9332
Generation: 870. 正答率(トレーニング) = 0.99
: 870. 正答率(テスト) = 0.9309
Generation: 880. 正答率(トレーニング) = 1.0
: 880. 正答率(テスト) = 0.9319
Generation: 890. 正答率(トレーニング) = 0.95
: 890. 正答率(テスト) = 0.9314
Generation: 900. 正答率(トレーニング) = 0.97
: 900. 正答率(テスト) = 0.9306
Generation: 910. 正答率(トレーニング) = 0.93
: 910. 正答率(テスト) = 0.9346
Generation: 920. 正答率(トレーニング) = 0.94
: 920. 正答率(テスト) = 0.9215
Generation: 930. 正答率(トレーニング) = 0.98
: 930. 正答率(テスト) = 0.9249
Generation: 940. 正答率(トレーニング) = 0.96
: 940. 正答率(テスト) = 0.9333
Generation: 950. 正答率(トレーニング) = 0.98
: 950. 正答率(テスト) = 0.9257
Generation: 960. 正答率(トレーニング) = 0.99
: 960. 正答率(テスト) = 0.9366
Generation: 970. 正答率(トレーニング) = 0.97
: 970. 正答率(テスト) = 0.9346
Generation: 980. 正答率(トレーニング) = 0.96
: 980. 正答率(テスト) = 0.9271
Generation: 990. 正答率(トレーニング) = 0.99
: 990. 正答率(テスト) = 0.9333
Generation: 1000. 正答率(トレーニング) = 0.94
: 1000. 正答率(テスト) = 0.9356



[try] 学習率を変えてみよう

RSMprop

```
In [26]: # データの読み込み
(x_train, d_train), (x_test, d_test) = load_mnist(normalize=True, one_hot_label=True)

print("データ読み込み完了")

# batch_normalizationの設定 =====
#use_batchnorm = True
use_batchnorm = False
# =====

network = MultiLayerNet(input_size=784, hidden_size_list=[40, 20], output_size=10, activation='sigmoid', weight_init_std=0.01,
                        use_batchnorm=use_batchnorm)

iters_num = 1000
train_size = x_train.shape[0]
batch_size = 100
learning_rate = 0.0001
decay_rate = 0.99

train_loss_list = []
accuracies_train = []
accuracies_test = []

plot_interval=10

for i in range(iters_num):
    batch_mask = np.random.choice(train_size, batch_size)
    x_batch = x_train[batch_mask]
    d_batch = d_train[batch_mask]

    # 勾配
    grad = network.gradient(x_batch, d_batch)
    if i == 0:
        h = {}
    for key in ('W1', 'W2', 'W3', 'b1', 'b2', 'b3'):
        if i == 0:
            h[key] = np.zeros_like(network.params[key])
        h[key] *= decay_rate
        h[key] += (1 - decay_rate) * np.square(grad[key])
        network.params[key] -= learning_rate * grad[key] / (np.sqrt(h[key]) + 1e-7)
```

```
    loss = network.loss(x_batch, d_batch)
    train_loss_list.append(loss)

    if (i + 1) % plot_interval == 0:
        accr_test = network.accuracy(x_test, d_test)
        accuracies_test.append(accr_test)
        accr_train = network.accuracy(x_batch, d_batch)
        accuracies_train.append(accr_train)

        print('Generation: ' + str(i+1) + '. 正答率(トレーニング) = ' + str(accr_train))
        print('                : ' + str(i+1) + '. 正答率(テスト) = ' + str(accr_test))

lists = range(0, iters_num, plot_interval)
plt.plot(lists, accuracies_train, label="training set")
plt.plot(lists, accuracies_test, label="test set")
plt.legend(loc="lower right")
plt.title("accuracy")
plt.xlabel("count")
plt.ylabel("accuracy")
plt.ylim(0, 1.0)
# グラフの表示
plt.show()
```

データ読み込み完了

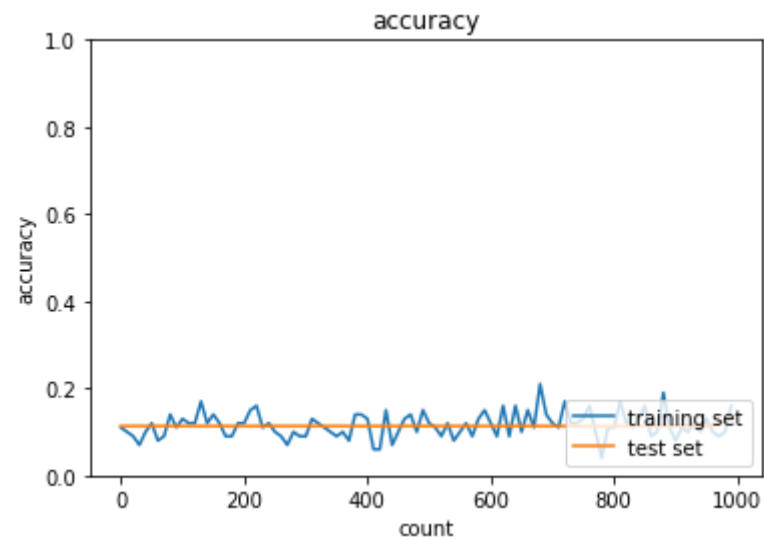
Generation: 10. 正答率(トレーニング) = 0.11
: 10. 正答率(テスト) = 0.1135
Generation: 20. 正答率(トレーニング) = 0.1
: 20. 正答率(テスト) = 0.1135
Generation: 30. 正答率(トレーニング) = 0.09
: 30. 正答率(テスト) = 0.1135
Generation: 40. 正答率(トレーニング) = 0.07
: 40. 正答率(テスト) = 0.1135
Generation: 50. 正答率(トレーニング) = 0.1
: 50. 正答率(テスト) = 0.1135
Generation: 60. 正答率(トレーニング) = 0.12
: 60. 正答率(テスト) = 0.1135
Generation: 70. 正答率(トレーニング) = 0.08
: 70. 正答率(テスト) = 0.1135
Generation: 80. 正答率(トレーニング) = 0.09
: 80. 正答率(テスト) = 0.1135
Generation: 90. 正答率(トレーニング) = 0.14
: 90. 正答率(テスト) = 0.1135
Generation: 100. 正答率(トレーニング) = 0.11
: 100. 正答率(テスト) = 0.1135
Generation: 110. 正答率(トレーニング) = 0.13
: 110. 正答率(テスト) = 0.1135
Generation: 120. 正答率(トレーニング) = 0.12
: 120. 正答率(テスト) = 0.1135
Generation: 130. 正答率(トレーニング) = 0.12
: 130. 正答率(テスト) = 0.1135
Generation: 140. 正答率(トレーニング) = 0.17
: 140. 正答率(テスト) = 0.1135
Generation: 150. 正答率(トレーニング) = 0.12
: 150. 正答率(テスト) = 0.1135
Generation: 160. 正答率(トレーニング) = 0.14
: 160. 正答率(テスト) = 0.1135
Generation: 170. 正答率(トレーニング) = 0.12
: 170. 正答率(テスト) = 0.1135
Generation: 180. 正答率(トレーニング) = 0.09
: 180. 正答率(テスト) = 0.1135
Generation: 190. 正答率(トレーニング) = 0.09
: 190. 正答率(テスト) = 0.1135
Generation: 200. 正答率(トレーニング) = 0.12
: 200. 正答率(テスト) = 0.1135

Generation: 210. 正答率(トレーニング) = 0.12
: 210. 正答率(テスト) = 0.1135
Generation: 220. 正答率(トレーニング) = 0.15
: 220. 正答率(テスト) = 0.1135
Generation: 230. 正答率(トレーニング) = 0.16
: 230. 正答率(テスト) = 0.1135
Generation: 240. 正答率(トレーニング) = 0.11
: 240. 正答率(テスト) = 0.1135
Generation: 250. 正答率(トレーニング) = 0.12
: 250. 正答率(テスト) = 0.1135
Generation: 260. 正答率(トレーニング) = 0.1
: 260. 正答率(テスト) = 0.1135
Generation: 270. 正答率(トレーニング) = 0.09
: 270. 正答率(テスト) = 0.1135
Generation: 280. 正答率(トレーニング) = 0.07
: 280. 正答率(テスト) = 0.1135
Generation: 290. 正答率(トレーニング) = 0.1
: 290. 正答率(テスト) = 0.1135
Generation: 300. 正答率(トレーニング) = 0.09
: 300. 正答率(テスト) = 0.1135
Generation: 310. 正答率(トレーニング) = 0.09
: 310. 正答率(テスト) = 0.1135
Generation: 320. 正答率(トレーニング) = 0.13
: 320. 正答率(テスト) = 0.1135
Generation: 330. 正答率(トレーニング) = 0.12
: 330. 正答率(テスト) = 0.1135
Generation: 340. 正答率(トレーニング) = 0.11
: 340. 正答率(テスト) = 0.1135
Generation: 350. 正答率(トレーニング) = 0.1
: 350. 正答率(テスト) = 0.1135
Generation: 360. 正答率(トレーニング) = 0.09
: 360. 正答率(テスト) = 0.1135
Generation: 370. 正答率(トレーニング) = 0.1
: 370. 正答率(テスト) = 0.1135
Generation: 380. 正答率(トレーニング) = 0.08
: 380. 正答率(テスト) = 0.1135
Generation: 390. 正答率(トレーニング) = 0.14
: 390. 正答率(テスト) = 0.1135
Generation: 400. 正答率(トレーニング) = 0.14
: 400. 正答率(テスト) = 0.1135
Generation: 410. 正答率(トレーニング) = 0.13
: 410. 正答率(テスト) = 0.1135

Generation: 420. 正答率(トレーニング) = 0.06
: 420. 正答率(テスト) = 0.1135
Generation: 430. 正答率(トレーニング) = 0.06
: 430. 正答率(テスト) = 0.1135
Generation: 440. 正答率(トレーニング) = 0.15
: 440. 正答率(テスト) = 0.1135
Generation: 450. 正答率(トレーニング) = 0.07
: 450. 正答率(テスト) = 0.1135
Generation: 460. 正答率(トレーニング) = 0.1
: 460. 正答率(テスト) = 0.1135
Generation: 470. 正答率(トレーニング) = 0.13
: 470. 正答率(テスト) = 0.1135
Generation: 480. 正答率(トレーニング) = 0.14
: 480. 正答率(テスト) = 0.1135
Generation: 490. 正答率(トレーニング) = 0.1
: 490. 正答率(テスト) = 0.1135
Generation: 500. 正答率(トレーニング) = 0.15
: 500. 正答率(テスト) = 0.1135
Generation: 510. 正答率(トレーニング) = 0.12
: 510. 正答率(テスト) = 0.1135
Generation: 520. 正答率(トレーニング) = 0.11
: 520. 正答率(テスト) = 0.1135
Generation: 530. 正答率(トレーニング) = 0.09
: 530. 正答率(テスト) = 0.1135
Generation: 540. 正答率(トレーニング) = 0.12
: 540. 正答率(テスト) = 0.1135
Generation: 550. 正答率(トレーニング) = 0.08
: 550. 正答率(テスト) = 0.1135
Generation: 560. 正答率(トレーニング) = 0.1
: 560. 正答率(テスト) = 0.1135
Generation: 570. 正答率(トレーニング) = 0.12
: 570. 正答率(テスト) = 0.1135
Generation: 580. 正答率(トレーニング) = 0.09
: 580. 正答率(テスト) = 0.1135
Generation: 590. 正答率(トレーニング) = 0.13
: 590. 正答率(テスト) = 0.1135
Generation: 600. 正答率(トレーニング) = 0.15
: 600. 正答率(テスト) = 0.1135
Generation: 610. 正答率(トレーニング) = 0.12
: 610. 正答率(テスト) = 0.1135
Generation: 620. 正答率(トレーニング) = 0.09
: 620. 正答率(テスト) = 0.1135

Generation: 630. 正答率(トレーニング) = 0.16
: 630. 正答率(テスト) = 0.1135
Generation: 640. 正答率(トレーニング) = 0.09
: 640. 正答率(テスト) = 0.1135
Generation: 650. 正答率(トレーニング) = 0.16
: 650. 正答率(テスト) = 0.1135
Generation: 660. 正答率(トレーニング) = 0.1
: 660. 正答率(テスト) = 0.1135
Generation: 670. 正答率(トレーニング) = 0.15
: 670. 正答率(テスト) = 0.1135
Generation: 680. 正答率(トレーニング) = 0.11
: 680. 正答率(テスト) = 0.1135
Generation: 690. 正答率(トレーニング) = 0.21
: 690. 正答率(テスト) = 0.1135
Generation: 700. 正答率(トレーニング) = 0.14
: 700. 正答率(テスト) = 0.1135
Generation: 710. 正答率(トレーニング) = 0.12
: 710. 正答率(テスト) = 0.1135
Generation: 720. 正答率(トレーニング) = 0.11
: 720. 正答率(テスト) = 0.1135
Generation: 730. 正答率(トレーニング) = 0.17
: 730. 正答率(テスト) = 0.1135
Generation: 740. 正答率(トレーニング) = 0.12
: 740. 正答率(テスト) = 0.1135
Generation: 750. 正答率(トレーニング) = 0.12
: 750. 正答率(テスト) = 0.1135
Generation: 760. 正答率(トレーニング) = 0.13
: 760. 正答率(テスト) = 0.1135
Generation: 770. 正答率(トレーニング) = 0.16
: 770. 正答率(テスト) = 0.1135
Generation: 780. 正答率(トレーニング) = 0.1
: 780. 正答率(テスト) = 0.1135
Generation: 790. 正答率(トレーニング) = 0.04
: 790. 正答率(テスト) = 0.1135
Generation: 800. 正答率(トレーニング) = 0.11
: 800. 正答率(テスト) = 0.1135
Generation: 810. 正答率(トレーニング) = 0.11
: 810. 正答率(テスト) = 0.1135
Generation: 820. 正答率(トレーニング) = 0.17
: 820. 正答率(テスト) = 0.1135
Generation: 830. 正答率(トレーニング) = 0.12
: 830. 正答率(テスト) = 0.1135

Generation: 840. 正答率(トレーニング) = 0.11
: 840. 正答率(テスト) = 0.1135
Generation: 850. 正答率(トレーニング) = 0.13
: 850. 正答率(テスト) = 0.1135
Generation: 860. 正答率(トレーニング) = 0.16
: 860. 正答率(テスト) = 0.1135
Generation: 870. 正答率(トレーニング) = 0.09
: 870. 正答率(テスト) = 0.1135
Generation: 880. 正答率(トレーニング) = 0.1
: 880. 正答率(テスト) = 0.1135
Generation: 890. 正答率(トレーニング) = 0.19
: 890. 正答率(テスト) = 0.1135
Generation: 900. 正答率(トレーニング) = 0.11
: 900. 正答率(テスト) = 0.1135
Generation: 910. 正答率(トレーニング) = 0.08
: 910. 正答率(テスト) = 0.1135
Generation: 920. 正答率(トレーニング) = 0.11
: 920. 正答率(テスト) = 0.1135
Generation: 930. 正答率(トレーニング) = 0.1
: 930. 正答率(テスト) = 0.1135
Generation: 940. 正答率(トレーニング) = 0.12
: 940. 正答率(テスト) = 0.1135
Generation: 950. 正答率(トレーニング) = 0.1
: 950. 正答率(テスト) = 0.1135
Generation: 960. 正答率(トレーニング) = 0.13
: 960. 正答率(テスト) = 0.1135
Generation: 970. 正答率(トレーニング) = 0.1
: 970. 正答率(テスト) = 0.1137
Generation: 980. 正答率(トレーニング) = 0.09
: 980. 正答率(テスト) = 0.1199
Generation: 990. 正答率(トレーニング) = 0.1
: 990. 正答率(テスト) = 0.1251
Generation: 1000. 正答率(トレーニング) = 0.16
: 1000. 正答率(テスト) = 0.1321



[try] 活性化関数と重みの初期化方法を変えてみよう

RSMprop

```
In [39]: # データの読み込み
(x_train, d_train), (x_test, d_test) = load_mnist(normalize=True, one_hot_label=True)

print("データ読み込み完了")

# batch_normalizationの設定 =====
#use_batchnorm = True
use_batchnorm = False
# =====

network = MultiLayerNet(input_size=784, hidden_size_list=[40, 20], output_size=10, activation='relu', weight_init_std='he',
                        use_batchnorm=use_batchnorm)

iters_num = 1000
train_size = x_train.shape[0]
batch_size = 100
learning_rate = 0.01
decay_rate = 0.99

train_loss_list = []
accuracies_train = []
accuracies_test = []

plot_interval=10

for i in range(iters_num):
    batch_mask = np.random.choice(train_size, batch_size)
    x_batch = x_train[batch_mask]
    d_batch = d_train[batch_mask]

    # 勾配
    grad = network.gradient(x_batch, d_batch)
    if i == 0:
        h = {}
    for key in ('W1', 'W2', 'W3', 'b1', 'b2', 'b3'):
        if i == 0:
            h[key] = np.zeros_like(network.params[key])
        h[key] *= decay_rate
        h[key] += (1 - decay_rate) * np.square(grad[key])
        network.params[key] -= learning_rate * grad[key] / (np.sqrt(h[key]) + 1e-7)
```

```
    loss = network.loss(x_batch, d_batch)
    train_loss_list.append(loss)

    if (i + 1) % plot_interval == 0:
        accr_test = network.accuracy(x_test, d_test)
        accuracies_test.append(accr_test)
        accr_train = network.accuracy(x_batch, d_batch)
        accuracies_train.append(accr_train)

        print('Generation: ' + str(i+1) + '. 正答率(トレーニング) = ' + str(accr_train))
        print('                : ' + str(i+1) + '. 正答率(テスト) = ' + str(accr_test))

lists = range(0, iters_num, plot_interval)
plt.plot(lists, accuracies_train, label="training set")
plt.plot(lists, accuracies_test, label="test set")
plt.legend(loc="lower right")
plt.title("accuracy")
plt.xlabel("count")
plt.ylabel("accuracy")
plt.ylim(0, 1.0)
# グラフの表示
plt.show()
```

データ読み込み完了

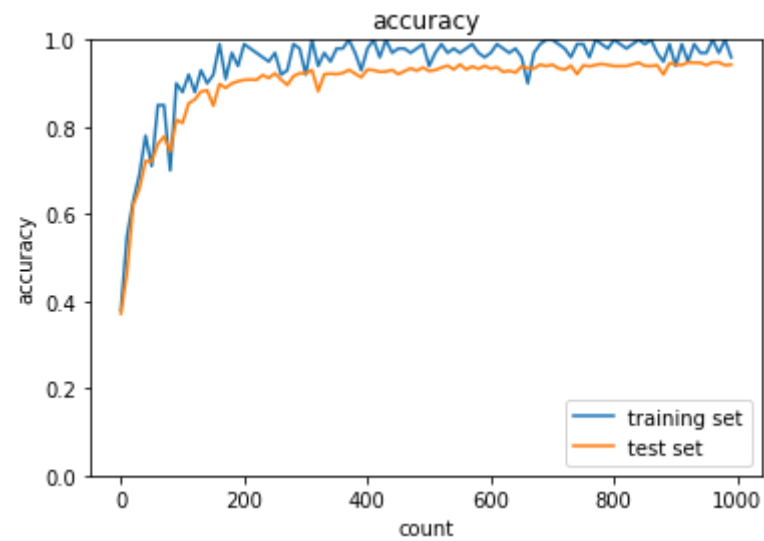
Generation: 10. 正答率(トレーニング) = 0.38
: 10. 正答率(テスト) = 0.3712
Generation: 20. 正答率(トレーニング) = 0.55
: 20. 正答率(テスト) = 0.4668
Generation: 30. 正答率(トレーニング) = 0.63
: 30. 正答率(テスト) = 0.6209
Generation: 40. 正答率(トレーニング) = 0.69
: 40. 正答率(テスト) = 0.658
Generation: 50. 正答率(トレーニング) = 0.78
: 50. 正答率(テスト) = 0.7215
Generation: 60. 正答率(トレーニング) = 0.71
: 60. 正答率(テスト) = 0.7212
Generation: 70. 正答率(トレーニング) = 0.85
: 70. 正答率(テスト) = 0.7607
Generation: 80. 正答率(トレーニング) = 0.85
: 80. 正答率(テスト) = 0.7781
Generation: 90. 正答率(トレーニング) = 0.7
: 90. 正答率(テスト) = 0.7435
Generation: 100. 正答率(トレーニング) = 0.9
: 100. 正答率(テスト) = 0.8162
Generation: 110. 正答率(トレーニング) = 0.88
: 110. 正答率(テスト) = 0.809
Generation: 120. 正答率(トレーニング) = 0.92
: 120. 正答率(テスト) = 0.8543
Generation: 130. 正答率(トレーニング) = 0.88
: 130. 正答率(テスト) = 0.863
Generation: 140. 正答率(トレーニング) = 0.93
: 140. 正答率(テスト) = 0.8811
Generation: 150. 正答率(トレーニング) = 0.9
: 150. 正答率(テスト) = 0.8842
Generation: 160. 正答率(トレーニング) = 0.92
: 160. 正答率(テスト) = 0.8486
Generation: 170. 正答率(トレーニング) = 0.99
: 170. 正答率(テスト) = 0.8984
Generation: 180. 正答率(トレーニング) = 0.91
: 180. 正答率(テスト) = 0.8895
Generation: 190. 正答率(トレーニング) = 0.97
: 190. 正答率(テスト) = 0.8999
Generation: 200. 正答率(トレーニング) = 0.94
: 200. 正答率(テスト) = 0.9047

Generation: 210. 正答率(トレーニング) = 0.99
: 210. 正答率(テスト) = 0.908
Generation: 220. 正答率(トレーニング) = 0.98
: 220. 正答率(テスト) = 0.9087
Generation: 230. 正答率(トレーニング) = 0.97
: 230. 正答率(テスト) = 0.9081
Generation: 240. 正答率(トレーニング) = 0.96
: 240. 正答率(テスト) = 0.9189
Generation: 250. 正答率(トレーニング) = 0.95
: 250. 正答率(テスト) = 0.9122
Generation: 260. 正答率(トレーニング) = 0.97
: 260. 正答率(テスト) = 0.923
Generation: 270. 正答率(トレーニング) = 0.92
: 270. 正答率(テスト) = 0.9084
Generation: 280. 正答率(トレーニング) = 0.93
: 280. 正答率(テスト) = 0.8962
Generation: 290. 正答率(トレーニング) = 0.99
: 290. 正答率(テスト) = 0.9167
Generation: 300. 正答率(トレーニング) = 0.98
: 300. 正答率(テスト) = 0.9231
Generation: 310. 正答率(トレーニング) = 0.92
: 310. 正答率(テスト) = 0.923
Generation: 320. 正答率(トレーニング) = 1.0
: 320. 正答率(テスト) = 0.9306
Generation: 330. 正答率(トレーニング) = 0.94
: 330. 正答率(テスト) = 0.8818
Generation: 340. 正答率(トレーニング) = 0.97
: 340. 正答率(テスト) = 0.9209
Generation: 350. 正答率(トレーニング) = 0.95
: 350. 正答率(テスト) = 0.9223
Generation: 360. 正答率(トレーニング) = 0.98
: 360. 正答率(テスト) = 0.9211
Generation: 370. 正答率(トレーニング) = 0.98
: 370. 正答率(テスト) = 0.9237
Generation: 380. 正答率(トレーニング) = 1.0
: 380. 正答率(テスト) = 0.9311
Generation: 390. 正答率(トレーニング) = 0.97
: 390. 正答率(テスト) = 0.9222
Generation: 400. 正答率(トレーニング) = 0.93
: 400. 正答率(テスト) = 0.9139
Generation: 410. 正答率(トレーニング) = 0.98
: 410. 正答率(テスト) = 0.9317

Generation: 420. 正答率(トレーニング) = 1.0
: 420. 正答率(テスト) = 0.9302
Generation: 430. 正答率(トレーニング) = 0.96
: 430. 正答率(テスト) = 0.9263
Generation: 440. 正答率(トレーニング) = 1.0
: 440. 正答率(テスト) = 0.9272
Generation: 450. 正答率(トレーニング) = 0.97
: 450. 正答率(テスト) = 0.9308
Generation: 460. 正答率(トレーニング) = 0.98
: 460. 正答率(テスト) = 0.9209
Generation: 470. 正答率(トレーニング) = 0.98
: 470. 正答率(テスト) = 0.9281
Generation: 480. 正答率(トレーニング) = 0.97
: 480. 正答率(テスト) = 0.9343
Generation: 490. 正答率(トレーニング) = 0.98
: 490. 正答率(テスト) = 0.929
Generation: 500. 正答率(トレーニング) = 0.99
: 500. 正答率(テスト) = 0.9365
Generation: 510. 正答率(トレーニング) = 0.94
: 510. 正答率(テスト) = 0.9284
Generation: 520. 正答率(トレーニング) = 0.97
: 520. 正答率(テスト) = 0.9306
Generation: 530. 正答率(トレーニング) = 0.99
: 530. 正答率(テスト) = 0.9362
Generation: 540. 正答率(トレーニング) = 0.97
: 540. 正答率(テスト) = 0.9407
Generation: 550. 正答率(トレーニング) = 0.98
: 550. 正答率(テスト) = 0.9318
Generation: 560. 正答率(トレーニング) = 0.97
: 560. 正答率(テスト) = 0.9435
Generation: 570. 正答率(トレーニング) = 0.98
: 570. 正答率(テスト) = 0.9319
Generation: 580. 正答率(トレーニング) = 0.99
: 580. 正答率(テスト) = 0.939
Generation: 590. 正答率(トレーニング) = 0.97
: 590. 正答率(テスト) = 0.9334
Generation: 600. 正答率(トレーニング) = 0.96
: 600. 正答率(テスト) = 0.9401
Generation: 610. 正答率(トレーニング) = 0.97
: 610. 正答率(テスト) = 0.9341
Generation: 620. 正答率(トレーニング) = 0.99
: 620. 正答率(テスト) = 0.937

Generation: 630. 正答率(トレーニング) = 0.98
: 630. 正答率(テスト) = 0.9262
Generation: 640. 正答率(トレーニング) = 0.97
: 640. 正答率(テスト) = 0.929
Generation: 650. 正答率(トレーニング) = 0.98
: 650. 正答率(テスト) = 0.9246
Generation: 660. 正答率(トレーニング) = 0.96
: 660. 正答率(テスト) = 0.9386
Generation: 670. 正答率(トレーニング) = 0.9
: 670. 正答率(テスト) = 0.9353
Generation: 680. 正答率(トレーニング) = 0.97
: 680. 正答率(テスト) = 0.9341
Generation: 690. 正答率(トレーニング) = 0.99
: 690. 正答率(テスト) = 0.943
Generation: 700. 正答率(トレーニング) = 1.0
: 700. 正答率(テスト) = 0.9397
Generation: 710. 正答率(トレーニング) = 1.0
: 710. 正答率(テスト) = 0.9429
Generation: 720. 正答率(トレーニング) = 0.99
: 720. 正答率(テスト) = 0.9345
Generation: 730. 正答率(トレーニング) = 0.98
: 730. 正答率(テスト) = 0.9315
Generation: 740. 正答率(トレーニング) = 0.96
: 740. 正答率(テスト) = 0.9406
Generation: 750. 正答率(トレーニング) = 0.99
: 750. 正答率(テスト) = 0.9212
Generation: 760. 正答率(トレーニング) = 0.99
: 760. 正答率(テスト) = 0.9411
Generation: 770. 正答率(トレーニング) = 0.96
: 770. 正答率(テスト) = 0.9392
Generation: 780. 正答率(トレーニング) = 1.0
: 780. 正答率(テスト) = 0.9422
Generation: 790. 正答率(トレーニング) = 0.99
: 790. 正答率(テスト) = 0.9447
Generation: 800. 正答率(トレーニング) = 0.98
: 800. 正答率(テスト) = 0.9422
Generation: 810. 正答率(トレーニング) = 1.0
: 810. 正答率(テスト) = 0.9397
Generation: 820. 正答率(トレーニング) = 0.99
: 820. 正答率(テスト) = 0.9397
Generation: 830. 正答率(トレーニング) = 0.98
: 830. 正答率(テスト) = 0.9397

Generation: 840. 正答率(トレーニング) = 0.99
: 840. 正答率(テスト) = 0.9437
Generation: 850. 正答率(トレーニング) = 1.0
: 850. 正答率(テスト) = 0.9474
Generation: 860. 正答率(トレーニング) = 0.99
: 860. 正答率(テスト) = 0.9403
Generation: 870. 正答率(トレーニング) = 1.0
: 870. 正答率(テスト) = 0.9398
Generation: 880. 正答率(トレーニング) = 0.97
: 880. 正答率(テスト) = 0.9416
Generation: 890. 正答率(トレーニング) = 0.95
: 890. 正答率(テスト) = 0.9204
Generation: 900. 正答率(トレーニング) = 0.99
: 900. 正答率(テスト) = 0.9462
Generation: 910. 正答率(トレーニング) = 0.94
: 910. 正答率(テスト) = 0.9434
Generation: 920. 正答率(トレーニング) = 0.99
: 920. 正答率(テスト) = 0.9421
Generation: 930. 正答率(トレーニング) = 0.95
: 930. 正答率(テスト) = 0.9483
Generation: 940. 正答率(トレーニング) = 0.99
: 940. 正答率(テスト) = 0.9474
Generation: 950. 正答率(トレーニング) = 0.97
: 950. 正答率(テスト) = 0.9473
Generation: 960. 正答率(トレーニング) = 0.97
: 960. 正答率(テスト) = 0.9417
Generation: 970. 正答率(トレーニング) = 1.0
: 970. 正答率(テスト) = 0.9484
Generation: 980. 正答率(トレーニング) = 0.97
: 980. 正答率(テスト) = 0.9483
Generation: 990. 正答率(トレーニング) = 1.0
: 990. 正答率(テスト) = 0.9416
Generation: 1000. 正答率(トレーニング) = 0.96
: 1000. 正答率(テスト) = 0.9429



[try] バッチ正規化をしてみよう & ReLU, He
RSMprop

```

In [51]: # データの読み込み
(x_train, d_train), (x_test, d_test) = load_mnist(normalize=True, one_hot_label=True)

print("データ読み込み完了")

# batch_normalizationの設定 =====
#use_batchnorm = True
use_batchnorm = False
# =====

network = MultiLayerNet(input_size=784, hidden_size_list=[40, 20], output_size=10, activation='relu', weight_init_std='he',
                        use_batchnorm=use_batchnorm)

iters_num = 1000
train_size = x_train.shape[0]
batch_size = 100
learning_rate = 0.01
decay_rate = 0.99

train_loss_list = []
accuracies_train = []
accuracies_test = []

plot_interval=10

for i in range(iters_num):
    batch_mask = np.random.choice(train_size, batch_size)
    x_batch = x_train[batch_mask]
    d_batch = d_train[batch_mask]

    # 勾配
    grad = network.gradient(x_batch, d_batch)
    if i == 0:
        h = {}
    for key in ('W1', 'W2', 'W3', 'b1', 'b2', 'b3'):
        if i == 0:
            h[key] = np.zeros_like(network.params[key])
        h[key] *= decay_rate
        h[key] += (1 - decay_rate) * np.square(grad[key])
        network.params[key] -= learning_rate * grad[key] / (np.sqrt(h[key]) + 1e-7)

```

```
    loss = network.loss(x_batch, d_batch)
    train_loss_list.append(loss)

    if (i + 1) % plot_interval == 0:
        accr_test = network.accuracy(x_test, d_test)
        accuracies_test.append(accr_test)
        accr_train = network.accuracy(x_batch, d_batch)
        accuracies_train.append(accr_train)

        print('Generation: ' + str(i+1) + '. 正答率(トレーニング) = ' + str(accr_train))
        print('                : ' + str(i+1) + '. 正答率(テスト) = ' + str(accr_test))

lists = range(0, iters_num, plot_interval)
plt.plot(lists, accuracies_train, label="training set")
plt.plot(lists, accuracies_test, label="test set")
plt.legend(loc="lower right")
plt.title("accuracy")
plt.xlabel("count")
plt.ylabel("accuracy")
plt.ylim(0, 1.0)
# グラフの表示
plt.show()
```

データ読み込み完了

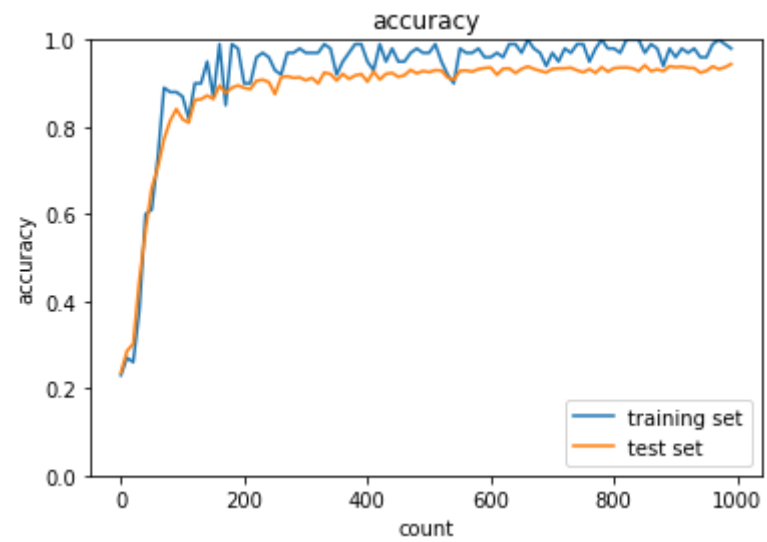
Generation: 10. 正答率(トレーニング) = 0.23
: 10. 正答率(テスト) = 0.2357
Generation: 20. 正答率(トレーニング) = 0.27
: 20. 正答率(テスト) = 0.286
Generation: 30. 正答率(トレーニング) = 0.26
: 30. 正答率(テスト) = 0.3017
Generation: 40. 正答率(トレーニング) = 0.38
: 40. 正答率(テスト) = 0.4495
Generation: 50. 正答率(トレーニング) = 0.6
: 50. 正答率(テスト) = 0.5613
Generation: 60. 正答率(トレーニング) = 0.61
: 60. 正答率(テスト) = 0.6597
Generation: 70. 正答率(トレーニング) = 0.73
: 70. 正答率(テスト) = 0.707
Generation: 80. 正答率(トレーニング) = 0.89
: 80. 正答率(テスト) = 0.7726
Generation: 90. 正答率(トレーニング) = 0.88
: 90. 正答率(テスト) = 0.8144
Generation: 100. 正答率(トレーニング) = 0.88
: 100. 正答率(テスト) = 0.8412
Generation: 110. 正答率(トレーニング) = 0.87
: 110. 正答率(テスト) = 0.8178
Generation: 120. 正答率(トレーニング) = 0.82
: 120. 正答率(テスト) = 0.8097
Generation: 130. 正答率(トレーニング) = 0.9
: 130. 正答率(テスト) = 0.8619
Generation: 140. 正答率(トレーニング) = 0.9
: 140. 正答率(テスト) = 0.8635
Generation: 150. 正答率(トレーニング) = 0.95
: 150. 正答率(テスト) = 0.8721
Generation: 160. 正答率(トレーニング) = 0.87
: 160. 正答率(テスト) = 0.8639
Generation: 170. 正答率(トレーニング) = 0.99
: 170. 正答率(テスト) = 0.8953
Generation: 180. 正答率(トレーニング) = 0.85
: 180. 正答率(テスト) = 0.8776
Generation: 190. 正答率(トレーニング) = 0.99
: 190. 正答率(テスト) = 0.8899
Generation: 200. 正答率(トレーニング) = 0.98
: 200. 正答率(テスト) = 0.8952

Generation: 210. 正答率(トレーニング) = 0.9
: 210. 正答率(テスト) = 0.8898
Generation: 220. 正答率(トレーニング) = 0.9
: 220. 正答率(テスト) = 0.8868
Generation: 230. 正答率(トレーニング) = 0.96
: 230. 正答率(テスト) = 0.9059
Generation: 240. 正答率(トレーニング) = 0.97
: 240. 正答率(テスト) = 0.9084
Generation: 250. 正答率(トレーニング) = 0.96
: 250. 正答率(テスト) = 0.903
Generation: 260. 正答率(トレーニング) = 0.93
: 260. 正答率(テスト) = 0.8752
Generation: 270. 正答率(トレーニング) = 0.92
: 270. 正答率(テスト) = 0.9141
Generation: 280. 正答率(トレーニング) = 0.97
: 280. 正答率(テスト) = 0.9163
Generation: 290. 正答率(トレーニング) = 0.97
: 290. 正答率(テスト) = 0.9125
Generation: 300. 正答率(トレーニング) = 0.98
: 300. 正答率(テスト) = 0.9138
Generation: 310. 正答率(トレーニング) = 0.97
: 310. 正答率(テスト) = 0.9071
Generation: 320. 正答率(トレーニング) = 0.97
: 320. 正答率(テスト) = 0.9126
Generation: 330. 正答率(トレーニング) = 0.97
: 330. 正答率(テスト) = 0.8998
Generation: 340. 正答率(トレーニング) = 0.99
: 340. 正答率(テスト) = 0.9248
Generation: 350. 正答率(トレーニング) = 0.98
: 350. 正答率(テスト) = 0.9214
Generation: 360. 正答率(トレーニング) = 0.92
: 360. 正答率(テスト) = 0.9064
Generation: 370. 正答率(トレーニング) = 0.95
: 370. 正答率(テスト) = 0.9219
Generation: 380. 正答率(トレーニング) = 0.97
: 380. 正答率(テスト) = 0.9102
Generation: 390. 正答率(トレーニング) = 0.99
: 390. 正答率(テスト) = 0.9181
Generation: 400. 正答率(トレーニング) = 0.99
: 400. 正答率(テスト) = 0.9214
Generation: 410. 正答率(トレーニング) = 0.95
: 410. 正答率(テスト) = 0.9039

Generation: 420. 正答率(トレーニング) = 0.93
: 420. 正答率(テスト) = 0.9273
Generation: 430. 正答率(トレーニング) = 0.99
: 430. 正答率(テスト) = 0.9089
Generation: 440. 正答率(トレーニング) = 0.95
: 440. 正答率(テスト) = 0.9218
Generation: 450. 正答率(トレーニング) = 0.98
: 450. 正答率(テスト) = 0.924
Generation: 460. 正答率(トレーニング) = 0.95
: 460. 正答率(テスト) = 0.9149
Generation: 470. 正答率(トレーニング) = 0.95
: 470. 正答率(テスト) = 0.9186
Generation: 480. 正答率(トレーニング) = 0.97
: 480. 正答率(テスト) = 0.931
Generation: 490. 正答率(トレーニング) = 0.98
: 490. 正答率(テスト) = 0.923
Generation: 500. 正答率(トレーニング) = 0.97
: 500. 正答率(テスト) = 0.9284
Generation: 510. 正答率(トレーニング) = 0.97
: 510. 正答率(テスト) = 0.9258
Generation: 520. 正答率(トレーニング) = 0.99
: 520. 正答率(テスト) = 0.9303
Generation: 530. 正答率(トレーニング) = 0.95
: 530. 正答率(テスト) = 0.9293
Generation: 540. 正答率(トレーニング) = 0.92
: 540. 正答率(テスト) = 0.9132
Generation: 550. 正答率(トレーニング) = 0.9
: 550. 正答率(テスト) = 0.9084
Generation: 560. 正答率(トレーニング) = 0.98
: 560. 正答率(テスト) = 0.9286
Generation: 570. 正答率(トレーニング) = 0.97
: 570. 正答率(テスト) = 0.9296
Generation: 580. 正答率(トレーニング) = 0.97
: 580. 正答率(テスト) = 0.9269
Generation: 590. 正答率(トレーニング) = 0.98
: 590. 正答率(テスト) = 0.9326
Generation: 600. 正答率(トレーニング) = 0.96
: 600. 正答率(テスト) = 0.9351
Generation: 610. 正答率(トレーニング) = 0.96
: 610. 正答率(テスト) = 0.9365
Generation: 620. 正答率(トレーニング) = 0.97
: 620. 正答率(テスト) = 0.9196

Generation: 630. 正答率(トレーニング) = 0.96
: 630. 正答率(テスト) = 0.9337
Generation: 640. 正答率(トレーニング) = 0.99
: 640. 正答率(テスト) = 0.9343
Generation: 650. 正答率(トレーニング) = 0.99
: 650. 正答率(テスト) = 0.9238
Generation: 660. 正答率(トレーニング) = 0.97
: 660. 正答率(テスト) = 0.9331
Generation: 670. 正答率(トレーニング) = 1.0
: 670. 正答率(テスト) = 0.9389
Generation: 680. 正答率(トレーニング) = 0.98
: 680. 正答率(テスト) = 0.9331
Generation: 690. 正答率(トレーニング) = 0.97
: 690. 正答率(テスト) = 0.9289
Generation: 700. 正答率(トレーニング) = 0.94
: 700. 正答率(テスト) = 0.9249
Generation: 710. 正答率(トレーニング) = 0.97
: 710. 正答率(テスト) = 0.9332
Generation: 720. 正答率(トレーニング) = 0.95
: 720. 正答率(テスト) = 0.9343
Generation: 730. 正答率(トレーニング) = 0.98
: 730. 正答率(テスト) = 0.9345
Generation: 740. 正答率(トレーニング) = 0.97
: 740. 正答率(テスト) = 0.9359
Generation: 750. 正答率(トレーニング) = 0.99
: 750. 正答率(テスト) = 0.9299
Generation: 760. 正答率(トレーニング) = 0.99
: 760. 正答率(テスト) = 0.9255
Generation: 770. 正答率(トレーニング) = 0.95
: 770. 正答率(テスト) = 0.9327
Generation: 780. 正答率(トレーニング) = 0.98
: 780. 正答率(テスト) = 0.9241
Generation: 790. 正答率(トレーニング) = 1.0
: 790. 正答率(テスト) = 0.9373
Generation: 800. 正答率(トレーニング) = 0.98
: 800. 正答率(テスト) = 0.9269
Generation: 810. 正答率(トレーニング) = 0.98
: 810. 正答率(テスト) = 0.9348
Generation: 820. 正答率(トレーニング) = 0.97
: 820. 正答率(テスト) = 0.9361
Generation: 830. 正答率(トレーニング) = 1.0
: 830. 正答率(テスト) = 0.9364

Generation: 840. 正答率(トレーニング) = 1.0
: 840. 正答率(テスト) = 0.9345
Generation: 850. 正答率(トレーニング) = 1.0
: 850. 正答率(テスト) = 0.9282
Generation: 860. 正答率(トレーニング) = 0.97
: 860. 正答率(テスト) = 0.9416
Generation: 870. 正答率(トレーニング) = 0.99
: 870. 正答率(テスト) = 0.928
Generation: 880. 正答率(トレーニング) = 0.98
: 880. 正答率(テスト) = 0.9323
Generation: 890. 正答率(トレーニング) = 0.94
: 890. 正答率(テスト) = 0.928
Generation: 900. 正答率(トレーニング) = 0.98
: 900. 正答率(テスト) = 0.9394
Generation: 910. 正答率(トレーニング) = 0.96
: 910. 正答率(テスト) = 0.937
Generation: 920. 正答率(トレーニング) = 0.98
: 920. 正答率(テスト) = 0.9383
Generation: 930. 正答率(トレーニング) = 0.97
: 930. 正答率(テスト) = 0.9357
Generation: 940. 正答率(トレーニング) = 0.98
: 940. 正答率(テスト) = 0.9351
Generation: 950. 正答率(トレーニング) = 0.96
: 950. 正答率(テスト) = 0.9246
Generation: 960. 正答率(トレーニング) = 0.96
: 960. 正答率(テスト) = 0.9288
Generation: 970. 正答率(トレーニング) = 0.99
: 970. 正答率(テスト) = 0.939
Generation: 980. 正答率(トレーニング) = 1.0
: 980. 正答率(テスト) = 0.9327
Generation: 990. 正答率(トレーニング) = 0.99
: 990. 正答率(テスト) = 0.9369
Generation: 1000. 正答率(トレーニング) = 0.98
: 1000. 正答率(テスト) = 0.9443



Adam

```
In [27]: # データの読み込み
(x_train, d_train), (x_test, d_test) = load_mnist(normalize=True, one_hot_label=True)

print("データ読み込み完了")

# batch_normalizationの設定 =====
#use_batchnorm = True
use_batchnorm = False
# =====

network = MultiLayerNet(input_size=784, hidden_size_list=[40, 20], output_size=10, activation='sigmoid', weight_init_std=0.01,
                        use_batchnorm=use_batchnorm)

iters_num = 1000
train_size = x_train.shape[0]
batch_size = 100
learning_rate = 0.01
beta1 = 0.9
beta2 = 0.999

train_loss_list = []
accuracies_train = []
accuracies_test = []

plot_interval=10

for i in range(iters_num):
    batch_mask = np.random.choice(train_size, batch_size)
    x_batch = x_train[batch_mask]
    d_batch = d_train[batch_mask]

    # 勾配
    grad = network.gradient(x_batch, d_batch)
    if i == 0:
        m = {}
        v = {}
    learning_rate_t = learning_rate * np.sqrt(1.0 - beta2 ** (i + 1)) / (1.0 - beta1 ** (i + 1))
    for key in ('W1', 'W2', 'W3', 'b1', 'b2', 'b3'):
        if i == 0:
            m[key] = np.zeros_like(network.params[key])
            v[key] = np.zeros_like(network.params[key])
```

```
m[key] += (1 - beta1) * (grad[key] - m[key])
v[key] += (1 - beta2) * (grad[key] ** 2 - v[key])
network.params[key] -= learning_rate_t * m[key] / (np.sqrt(v[key]) + 1e-7)

if (i + 1) % plot_interval == 0:
    accr_test = network.accuracy(x_test, d_test)
    accuracies_test.append(accr_test)
    accr_train = network.accuracy(x_batch, d_batch)
    accuracies_train.append(accr_train)
    loss = network.loss(x_batch, d_batch)
    train_loss_list.append(loss)

    print(' Generation: ' + str(i+1) + '. 正答率(トレーニング) = ' + str(accr_train))
    print('           : ' + str(i+1) + '. 正答率(テスト) = ' + str(accr_test))

lists = range(0, iters_num, plot_interval)
plt.plot(lists, accuracies_train, label="training set")
plt.plot(lists, accuracies_test, label="test set")
plt.legend(loc="lower right")
plt.title("accuracy")
plt.xlabel("count")
plt.ylabel("accuracy")
plt.ylim(0, 1.0)
# グラフの表示
plt.show()
```

データ読み込み完了

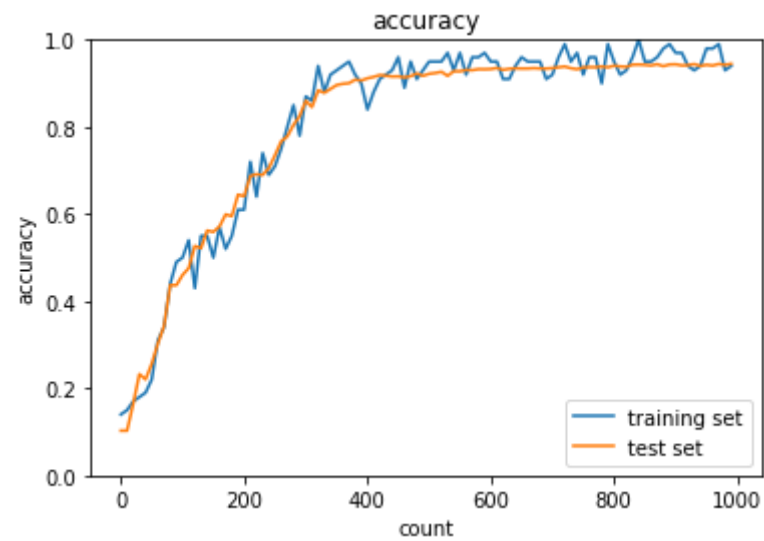
Generation: 10. 正答率(トレーニング) = 0.14
: 10. 正答率(テスト) = 0.1028
Generation: 20. 正答率(トレーニング) = 0.15
: 20. 正答率(テスト) = 0.1028
Generation: 30. 正答率(トレーニング) = 0.17
: 30. 正答率(テスト) = 0.1646
Generation: 40. 正答率(トレーニング) = 0.18
: 40. 正答率(テスト) = 0.2326
Generation: 50. 正答率(トレーニング) = 0.19
: 50. 正答率(テスト) = 0.2205
Generation: 60. 正答率(トレーニング) = 0.22
: 60. 正答率(テスト) = 0.2561
Generation: 70. 正答率(トレーニング) = 0.31
: 70. 正答率(テスト) = 0.3027
Generation: 80. 正答率(トレーニング) = 0.34
: 80. 正答率(テスト) = 0.343
Generation: 90. 正答率(トレーニング) = 0.44
: 90. 正答率(テスト) = 0.4369
Generation: 100. 正答率(トレーニング) = 0.49
: 100. 正答率(テスト) = 0.4366
Generation: 110. 正答率(トレーニング) = 0.5
: 110. 正答率(テスト) = 0.4601
Generation: 120. 正答率(トレーニング) = 0.54
: 120. 正答率(テスト) = 0.4757
Generation: 130. 正答率(トレーニング) = 0.43
: 130. 正答率(テスト) = 0.5267
Generation: 140. 正答率(トレーニング) = 0.55
: 140. 正答率(テスト) = 0.5216
Generation: 150. 正答率(トレーニング) = 0.55
: 150. 正答率(テスト) = 0.5621
Generation: 160. 正答率(トレーニング) = 0.5
: 160. 正答率(テスト) = 0.559
Generation: 170. 正答率(トレーニング) = 0.57
: 170. 正答率(テスト) = 0.5715
Generation: 180. 正答率(トレーニング) = 0.52
: 180. 正答率(テスト) = 0.5992
Generation: 190. 正答率(トレーニング) = 0.55
: 190. 正答率(テスト) = 0.5952
Generation: 200. 正答率(トレーニング) = 0.61
: 200. 正答率(テスト) = 0.6445

Generation: 210. 正答率(トレーニング) = 0.61
: 210. 正答率(テスト) = 0.6411
Generation: 220. 正答率(トレーニング) = 0.72
: 220. 正答率(テスト) = 0.6874
Generation: 230. 正答率(トレーニング) = 0.64
: 230. 正答率(テスト) = 0.6914
Generation: 240. 正答率(トレーニング) = 0.74
: 240. 正答率(テスト) = 0.6898
Generation: 250. 正答率(トレーニング) = 0.69
: 250. 正答率(テスト) = 0.7053
Generation: 260. 正答率(トレーニング) = 0.71
: 260. 正答率(テスト) = 0.735
Generation: 270. 正答率(トレーニング) = 0.75
: 270. 正答率(テスト) = 0.7669
Generation: 280. 正答率(トレーニング) = 0.8
: 280. 正答率(テスト) = 0.7782
Generation: 290. 正答率(トレーニング) = 0.85
: 290. 正答率(テスト) = 0.8046
Generation: 300. 正答率(トレーニング) = 0.78
: 300. 正答率(テスト) = 0.8248
Generation: 310. 正答率(トレーニング) = 0.87
: 310. 正答率(テスト) = 0.8602
Generation: 320. 正答率(トレーニング) = 0.86
: 320. 正答率(テスト) = 0.8456
Generation: 330. 正答率(トレーニング) = 0.94
: 330. 正答率(テスト) = 0.8838
Generation: 340. 正答率(トレーニング) = 0.88
: 340. 正答率(テスト) = 0.8784
Generation: 350. 正答率(トレーニング) = 0.92
: 350. 正答率(テスト) = 0.8872
Generation: 360. 正答率(トレーニング) = 0.93
: 360. 正答率(テスト) = 0.8961
Generation: 370. 正答率(トレーニング) = 0.94
: 370. 正答率(テスト) = 0.899
Generation: 380. 正答率(トレーニング) = 0.95
: 380. 正答率(テスト) = 0.9004
Generation: 390. 正答率(トレーニング) = 0.92
: 390. 正答率(テスト) = 0.9084
Generation: 400. 正答率(トレーニング) = 0.9
: 400. 正答率(テスト) = 0.9066
Generation: 410. 正答率(トレーニング) = 0.84
: 410. 正答率(テスト) = 0.9116

Generation: 420. 正答率(トレーニング) = 0.88
: 420. 正答率(テスト) = 0.9148
Generation: 430. 正答率(トレーニング) = 0.91
: 430. 正答率(テスト) = 0.9191
Generation: 440. 正答率(トレーニング) = 0.92
: 440. 正答率(テスト) = 0.9175
Generation: 450. 正答率(トレーニング) = 0.93
: 450. 正答率(テスト) = 0.9157
Generation: 460. 正答率(トレーニング) = 0.96
: 460. 正答率(テスト) = 0.916
Generation: 470. 正答率(トレーニング) = 0.89
: 470. 正答率(テスト) = 0.9136
Generation: 480. 正答率(トレーニング) = 0.95
: 480. 正答率(テスト) = 0.9165
Generation: 490. 正答率(トレーニング) = 0.91
: 490. 正答率(テスト) = 0.9228
Generation: 500. 正答率(トレーニング) = 0.93
: 500. 正答率(テスト) = 0.9173
Generation: 510. 正答率(トレーニング) = 0.95
: 510. 正答率(テスト) = 0.922
Generation: 520. 正答率(トレーニング) = 0.95
: 520. 正答率(テスト) = 0.9237
Generation: 530. 正答率(トレーニング) = 0.95
: 530. 正答率(テスト) = 0.9266
Generation: 540. 正答率(トレーニング) = 0.97
: 540. 正答率(テスト) = 0.9172
Generation: 550. 正答率(トレーニング) = 0.93
: 550. 正答率(テスト) = 0.9279
Generation: 560. 正答率(トレーニング) = 0.97
: 560. 正答率(テスト) = 0.9277
Generation: 570. 正答率(トレーニング) = 0.92
: 570. 正答率(テスト) = 0.9307
Generation: 580. 正答率(トレーニング) = 0.96
: 580. 正答率(テスト) = 0.9303
Generation: 590. 正答率(トレーニング) = 0.96
: 590. 正答率(テスト) = 0.9326
Generation: 600. 正答率(トレーニング) = 0.97
: 600. 正答率(テスト) = 0.9324
Generation: 610. 正答率(トレーニング) = 0.95
: 610. 正答率(テスト) = 0.9329
Generation: 620. 正答率(トレーニング) = 0.95
: 620. 正答率(テスト) = 0.9353

Generation: 630. 正答率(トレーニング) = 0.91
: 630. 正答率(テスト) = 0.9311
Generation: 640. 正答率(トレーニング) = 0.91
: 640. 正答率(テスト) = 0.9334
Generation: 650. 正答率(トレーニング) = 0.94
: 650. 正答率(テスト) = 0.9345
Generation: 660. 正答率(トレーニング) = 0.96
: 660. 正答率(テスト) = 0.934
Generation: 670. 正答率(トレーニング) = 0.95
: 670. 正答率(テスト) = 0.9341
Generation: 680. 正答率(トレーニング) = 0.95
: 680. 正答率(テスト) = 0.9349
Generation: 690. 正答率(トレーニング) = 0.95
: 690. 正答率(テスト) = 0.9341
Generation: 700. 正答率(トレーニング) = 0.91
: 700. 正答率(テスト) = 0.9339
Generation: 710. 正答率(トレーニング) = 0.92
: 710. 正答率(テスト) = 0.9351
Generation: 720. 正答率(トレーニング) = 0.96
: 720. 正答率(テスト) = 0.9369
Generation: 730. 正答率(トレーニング) = 0.99
: 730. 正答率(テスト) = 0.9386
Generation: 740. 正答率(トレーニング) = 0.95
: 740. 正答率(テスト) = 0.9348
Generation: 750. 正答率(トレーニング) = 0.97
: 750. 正答率(テスト) = 0.9325
Generation: 760. 正答率(トレーニング) = 0.92
: 760. 正答率(テスト) = 0.9364
Generation: 770. 正答率(トレーニング) = 0.96
: 770. 正答率(テスト) = 0.9373
Generation: 780. 正答率(トレーニング) = 0.96
: 780. 正答率(テスト) = 0.9371
Generation: 790. 正答率(トレーニング) = 0.9
: 790. 正答率(テスト) = 0.9386
Generation: 800. 正答率(トレーニング) = 0.99
: 800. 正答率(テスト) = 0.9369
Generation: 810. 正答率(トレーニング) = 0.95
: 810. 正答率(テスト) = 0.9405
Generation: 820. 正答率(トレーニング) = 0.92
: 820. 正答率(テスト) = 0.9388
Generation: 830. 正答率(トレーニング) = 0.93
: 830. 正答率(テスト) = 0.9378

Generation: 840. 正答率(トレーニング) = 0.96
: 840. 正答率(テスト) = 0.9428
Generation: 850. 正答率(トレーニング) = 1.0
: 850. 正答率(テスト) = 0.9432
Generation: 860. 正答率(トレーニング) = 0.95
: 860. 正答率(テスト) = 0.9425
Generation: 870. 正答率(トレーニング) = 0.95
: 870. 正答率(テスト) = 0.9411
Generation: 880. 正答率(トレーニング) = 0.96
: 880. 正答率(テスト) = 0.9439
Generation: 890. 正答率(トレーニング) = 0.98
: 890. 正答率(テスト) = 0.9395
Generation: 900. 正答率(トレーニング) = 0.99
: 900. 正答率(テスト) = 0.9433
Generation: 910. 正答率(トレーニング) = 0.97
: 910. 正答率(テスト) = 0.9437
Generation: 920. 正答率(トレーニング) = 0.97
: 920. 正答率(テスト) = 0.9412
Generation: 930. 正答率(トレーニング) = 0.94
: 930. 正答率(テスト) = 0.9418
Generation: 940. 正答率(トレーニング) = 0.93
: 940. 正答率(テスト) = 0.9442
Generation: 950. 正答率(トレーニング) = 0.94
: 950. 正答率(テスト) = 0.9399
Generation: 960. 正答率(トレーニング) = 0.98
: 960. 正答率(テスト) = 0.9426
Generation: 970. 正答率(トレーニング) = 0.98
: 970. 正答率(テスト) = 0.9409
Generation: 980. 正答率(トレーニング) = 0.99
: 980. 正答率(テスト) = 0.9446
Generation: 990. 正答率(トレーニング) = 0.93
: 990. 正答率(テスト) = 0.9423
Generation: 1000. 正答率(トレーニング) = 0.94
: 1000. 正答率(テスト) = 0.9449



[try] 学習率を変えてみよう

Adam

```
In [30]: # データの読み込み
(x_train, d_train), (x_test, d_test) = load_mnist(normalize=True, one_hot_label=True)

print("データ読み込み完了")

# batch_normalizationの設定 =====
#use_batchnorm = True
use_batchnorm = False
# =====

network = MultiLayerNet(input_size=784, hidden_size_list=[40, 20], output_size=10, activation='sigmoid', weight_init_std=0.01,
                        use_batchnorm=use_batchnorm)

iters_num = 1000
train_size = x_train.shape[0]
batch_size = 100
learning_rate = 1
beta1 = 0.9
beta2 = 0.999

train_loss_list = []
accuracies_train = []
accuracies_test = []

plot_interval=10

for i in range(iters_num):
    batch_mask = np.random.choice(train_size, batch_size)
    x_batch = x_train[batch_mask]
    d_batch = d_train[batch_mask]

    # 勾配
    grad = network.gradient(x_batch, d_batch)
    if i == 0:
        m = {}
        v = {}
    learning_rate_t = learning_rate * np.sqrt(1.0 - beta2 ** (i + 1)) / (1.0 - beta1 ** (i + 1))
    for key in ('W1', 'W2', 'W3', 'b1', 'b2', 'b3'):
        if i == 0:
            m[key] = np.zeros_like(network.params[key])
            v[key] = np.zeros_like(network.params[key])
```

```
m[key] += (1 - beta1) * (grad[key] - m[key])
v[key] += (1 - beta2) * (grad[key] ** 2 - v[key])
network.params[key] -= learning_rate_t * m[key] / (np.sqrt(v[key]) + 1e-7)

if (i + 1) % plot_interval == 0:
    accr_test = network.accuracy(x_test, d_test)
    accuracies_test.append(accr_test)
    accr_train = network.accuracy(x_batch, d_batch)
    accuracies_train.append(accr_train)
    loss = network.loss(x_batch, d_batch)
    train_loss_list.append(loss)

    print('Generation: ' + str(i+1) + '. 正答率(トレーニング) = ' + str(accr_train))
    print('          : ' + str(i+1) + '. 正答率(テスト) = ' + str(accr_test))

lists = range(0, iters_num, plot_interval)
plt.plot(lists, accuracies_train, label="training set")
plt.plot(lists, accuracies_test, label="test set")
plt.legend(loc="lower right")
plt.title("accuracy")
plt.xlabel("count")
plt.ylabel("accuracy")
plt.ylim(0, 1.0)
# グラフの表示
plt.show()
```

データ読み込み完了

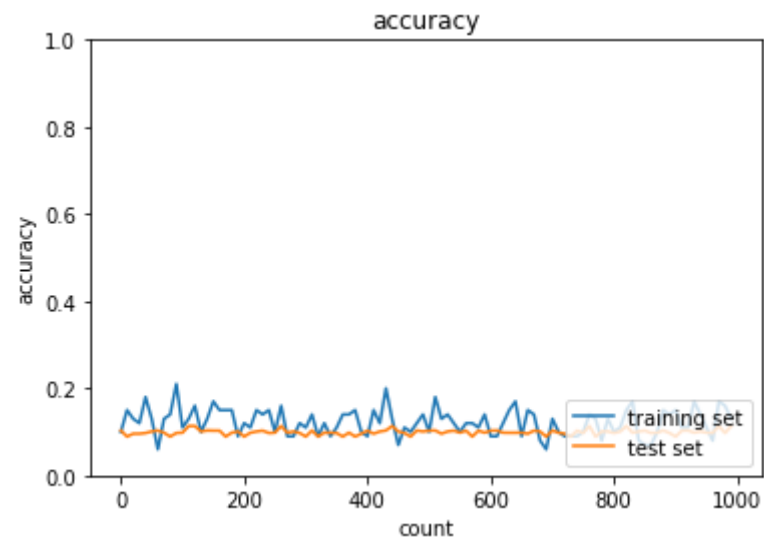
Generation: 10. 正答率(トレーニング) = 0.1
: 10. 正答率(テスト) = 0.1032
Generation: 20. 正答率(トレーニング) = 0.15
: 20. 正答率(テスト) = 0.0892
Generation: 30. 正答率(トレーニング) = 0.13
: 30. 正答率(テスト) = 0.0958
Generation: 40. 正答率(トレーニング) = 0.12
: 40. 正答率(テスト) = 0.0958
Generation: 50. 正答率(トレーニング) = 0.18
: 50. 正答率(テスト) = 0.0974
Generation: 60. 正答率(トレーニング) = 0.13
: 60. 正答率(テスト) = 0.101
Generation: 70. 正答率(トレーニング) = 0.06
: 70. 正答率(テスト) = 0.1032
Generation: 80. 正答率(トレーニング) = 0.13
: 80. 正答率(テスト) = 0.098
Generation: 90. 正答率(トレーニング) = 0.14
: 90. 正答率(テスト) = 0.0892
Generation: 100. 正答率(トレーニング) = 0.21
: 100. 正答率(テスト) = 0.0974
Generation: 110. 正答率(トレーニング) = 0.11
: 110. 正答率(テスト) = 0.0982
Generation: 120. 正答率(トレーニング) = 0.13
: 120. 正答率(テスト) = 0.1135
Generation: 130. 正答率(トレーニング) = 0.16
: 130. 正答率(テスト) = 0.1135
Generation: 140. 正答率(トレーニング) = 0.1
: 140. 正答率(テスト) = 0.1009
Generation: 150. 正答率(トレーニング) = 0.13
: 150. 正答率(テスト) = 0.1032
Generation: 160. 正答率(トレーニング) = 0.17
: 160. 正答率(テスト) = 0.1028
Generation: 170. 正答率(トレーニング) = 0.15
: 170. 正答率(テスト) = 0.1028
Generation: 180. 正答率(トレーニング) = 0.15
: 180. 正答率(テスト) = 0.0892
Generation: 190. 正答率(トレーニング) = 0.15
: 190. 正答率(テスト) = 0.0982
Generation: 200. 正答率(トレーニング) = 0.09
: 200. 正答率(テスト) = 0.101

Generation: 210. 正答率(トレーニング) = 0.12
: 210. 正答率(テスト) = 0.0892
Generation: 220. 正答率(トレーニング) = 0.11
: 220. 正答率(テスト) = 0.098
Generation: 230. 正答率(トレーニング) = 0.15
: 230. 正答率(テスト) = 0.1009
Generation: 240. 正答率(トレーニング) = 0.14
: 240. 正答率(テスト) = 0.1028
Generation: 250. 正答率(トレーニング) = 0.15
: 250. 正答率(テスト) = 0.0974
Generation: 260. 正答率(トレーニング) = 0.1
: 260. 正答率(テスト) = 0.0982
Generation: 270. 正答率(トレーニング) = 0.16
: 270. 正答率(テスト) = 0.1135
Generation: 280. 正答率(トレーニング) = 0.09
: 280. 正答率(テスト) = 0.098
Generation: 290. 正答率(トレーニング) = 0.09
: 290. 正答率(テスト) = 0.1009
Generation: 300. 正答率(トレーニング) = 0.12
: 300. 正答率(テスト) = 0.0974
Generation: 310. 正答率(トレーニング) = 0.11
: 310. 正答率(テスト) = 0.0892
Generation: 320. 正答率(トレーニング) = 0.14
: 320. 正答率(テスト) = 0.1032
Generation: 330. 正答率(トレーニング) = 0.09
: 330. 正答率(テスト) = 0.0892
Generation: 340. 正答率(トレーニング) = 0.12
: 340. 正答率(テスト) = 0.0982
Generation: 350. 正答率(トレーニング) = 0.09
: 350. 正答率(テスト) = 0.098
Generation: 360. 正答率(トレーニング) = 0.11
: 360. 正答率(テスト) = 0.0974
Generation: 370. 正答率(トレーニング) = 0.14
: 370. 正答率(テスト) = 0.0892
Generation: 380. 正答率(トレーニング) = 0.14
: 380. 正答率(テスト) = 0.0982
Generation: 390. 正答率(トレーニング) = 0.15
: 390. 正答率(テスト) = 0.0892
Generation: 400. 正答率(トレーニング) = 0.1
: 400. 正答率(テスト) = 0.0958
Generation: 410. 正答率(トレーニング) = 0.09
: 410. 正答率(テスト) = 0.1028

Generation: 420. 正答率(トレーニング) = 0.15
: 420. 正答率(テスト) = 0.0958
Generation: 430. 正答率(トレーニング) = 0.12
: 430. 正答率(テスト) = 0.1009
Generation: 440. 正答率(トレーニング) = 0.2
: 440. 正答率(テスト) = 0.1032
Generation: 450. 正答率(トレーニング) = 0.13
: 450. 正答率(テスト) = 0.1135
Generation: 460. 正答率(トレーニング) = 0.07
: 460. 正答率(テスト) = 0.1009
Generation: 470. 正答率(トレーニング) = 0.11
: 470. 正答率(テスト) = 0.0974
Generation: 480. 正答率(トレーニング) = 0.1
: 480. 正答率(テスト) = 0.0892
Generation: 490. 正答率(トレーニング) = 0.12
: 490. 正答率(テスト) = 0.1032
Generation: 500. 正答率(トレーニング) = 0.14
: 500. 正答率(テスト) = 0.1009
Generation: 510. 正答率(トレーニング) = 0.1
: 510. 正答率(テスト) = 0.1028
Generation: 520. 正答率(トレーニング) = 0.18
: 520. 正答率(テスト) = 0.1032
Generation: 530. 正答率(トレーニング) = 0.13
: 530. 正答率(テスト) = 0.0958
Generation: 540. 正答率(トレーニング) = 0.14
: 540. 正答率(テスト) = 0.1009
Generation: 550. 正答率(トレーニング) = 0.12
: 550. 正答率(テスト) = 0.1028
Generation: 560. 正答率(トレーニング) = 0.1
: 560. 正答率(テスト) = 0.0982
Generation: 570. 正答率(トレーニング) = 0.12
: 570. 正答率(テスト) = 0.1028
Generation: 580. 正答率(トレーニング) = 0.12
: 580. 正答率(テスト) = 0.0892
Generation: 590. 正答率(トレーニング) = 0.11
: 590. 正答率(テスト) = 0.1028
Generation: 600. 正答率(トレーニング) = 0.14
: 600. 正答率(テスト) = 0.0982
Generation: 610. 正答率(トレーニング) = 0.09
: 610. 正答率(テスト) = 0.1028
Generation: 620. 正答率(トレーニング) = 0.09
: 620. 正答率(テスト) = 0.1032

Generation: 630. 正答率(トレーニング) = 0.12
: 630. 正答率(テスト) = 0.0982
Generation: 640. 正答率(トレーニング) = 0.15
: 640. 正答率(テスト) = 0.098
Generation: 650. 正答率(トレーニング) = 0.17
: 650. 正答率(テスト) = 0.098
Generation: 660. 正答率(トレーニング) = 0.09
: 660. 正答率(テスト) = 0.0982
Generation: 670. 正答率(トレーニング) = 0.15
: 670. 正答率(テスト) = 0.0958
Generation: 680. 正答率(トレーニング) = 0.14
: 680. 正答率(テスト) = 0.1028
Generation: 690. 正答率(トレーニング) = 0.08
: 690. 正答率(テスト) = 0.1009
Generation: 700. 正答率(トレーニング) = 0.06
: 700. 正答率(テスト) = 0.0892
Generation: 710. 正答率(トレーニング) = 0.13
: 710. 正答率(テスト) = 0.1032
Generation: 720. 正答率(トレーニング) = 0.1
: 720. 正答率(テスト) = 0.098
Generation: 730. 正答率(トレーニング) = 0.09
: 730. 正答率(テスト) = 0.0958
Generation: 740. 正答率(トレーニング) = 0.09
: 740. 正答率(テスト) = 0.0892
Generation: 750. 正答率(トレーニング) = 0.09
: 750. 正答率(テスト) = 0.1028
Generation: 760. 正答率(トレーニング) = 0.1
: 760. 正答率(テスト) = 0.0974
Generation: 770. 正答率(トレーニング) = 0.14
: 770. 正答率(テスト) = 0.1135
Generation: 780. 正答率(トレーニング) = 0.13
: 780. 正答率(テスト) = 0.0892
Generation: 790. 正答率(トレーニング) = 0.08
: 790. 正答率(テスト) = 0.101
Generation: 800. 正答率(トレーニング) = 0.13
: 800. 正答率(テスト) = 0.101
Generation: 810. 正答率(トレーニング) = 0.1
: 810. 正答率(テスト) = 0.0974
Generation: 820. 正答率(トレーニング) = 0.11
: 820. 正答率(テスト) = 0.1009
Generation: 830. 正答率(トレーニング) = 0.15
: 830. 正答率(テスト) = 0.1135

Generation: 840. 正答率(トレーニング) = 0.17
: 840. 正答率(テスト) = 0.098
Generation: 850. 正答率(トレーニング) = 0.08
: 850. 正答率(テスト) = 0.1009
Generation: 860. 正答率(トレーニング) = 0.07
: 860. 正答率(テスト) = 0.1028
Generation: 870. 正答率(トレーニング) = 0.07
: 870. 正答率(テスト) = 0.0974
Generation: 880. 正答率(トレーニング) = 0.1
: 880. 正答率(テスト) = 0.098
Generation: 890. 正答率(トレーニング) = 0.15
: 890. 正答率(テスト) = 0.1032
Generation: 900. 正答率(トレーニング) = 0.14
: 900. 正答率(テスト) = 0.0974
Generation: 910. 正答率(トレーニング) = 0.15
: 910. 正答率(テスト) = 0.0892
Generation: 920. 正答率(トレーニング) = 0.11
: 920. 正答率(テスト) = 0.1028
Generation: 930. 正答率(トレーニング) = 0.11
: 930. 正答率(テスト) = 0.1032
Generation: 940. 正答率(トレーニング) = 0.17
: 940. 正答率(テスト) = 0.0982
Generation: 950. 正答率(トレーニング) = 0.14
: 950. 正答率(テスト) = 0.101
Generation: 960. 正答率(トレーニング) = 0.11
: 960. 正答率(テスト) = 0.0982
Generation: 970. 正答率(トレーニング) = 0.08
: 970. 正答率(テスト) = 0.0974
Generation: 980. 正答率(トレーニング) = 0.17
: 980. 正答率(テスト) = 0.1135
Generation: 990. 正答率(トレーニング) = 0.16
: 990. 正答率(テスト) = 0.0974
Generation: 1000. 正答率(トレーニング) = 0.13
: 1000. 正答率(テスト) = 0.1135



[try] 活性化関数と重みの初期化方法を変えてみよう

Adam

```

In [42]: # データの読み込み
(x_train, d_train), (x_test, d_test) = load_mnist(normalize=True, one_hot_label=True)

print("データ読み込み完了")

# batch_normalizationの設定 =====
#use_batchnorm = True
use_batchnorm = False
# =====

network = MultiLayerNet(input_size=784, hidden_size_list=[40, 20], output_size=10, activation='relu', weight_init_std='he',
                        use_batchnorm=use_batchnorm)

iters_num = 1000
train_size = x_train.shape[0]
batch_size = 100
learning_rate = 0.01
beta1 = 0.9
beta2 = 0.999

train_loss_list = []
accuracies_train = []
accuracies_test = []

plot_interval=10

for i in range(iters_num):
    batch_mask = np.random.choice(train_size, batch_size)
    x_batch = x_train[batch_mask]
    d_batch = d_train[batch_mask]

    # 勾配
    grad = network.gradient(x_batch, d_batch)
    if i == 0:
        m = {}
        v = {}
    learning_rate_t = learning_rate * np.sqrt(1.0 - beta2 ** (i + 1)) / (1.0 - beta1 ** (i + 1))
    for key in ('W1', 'W2', 'W3', 'b1', 'b2', 'b3'):
        if i == 0:
            m[key] = np.zeros_like(network.params[key])
            v[key] = np.zeros_like(network.params[key])

```

```
m[key] += (1 - beta1) * (grad[key] - m[key])
v[key] += (1 - beta2) * (grad[key] ** 2 - v[key])
network.params[key] -= learning_rate_t * m[key] / (np.sqrt(v[key]) + 1e-7)

if (i + 1) % plot_interval == 0:
    accr_test = network.accuracy(x_test, d_test)
    accuracies_test.append(accr_test)
    accr_train = network.accuracy(x_batch, d_batch)
    accuracies_train.append(accr_train)
    loss = network.loss(x_batch, d_batch)
    train_loss_list.append(loss)

    print('Generation: ' + str(i+1) + '. 正答率(トレーニング) = ' + str(accr_train))
    print('          : ' + str(i+1) + '. 正答率(テスト) = ' + str(accr_test))

lists = range(0, iters_num, plot_interval)
plt.plot(lists, accuracies_train, label="training set")
plt.plot(lists, accuracies_test, label="test set")
plt.legend(loc="lower right")
plt.title("accuracy")
plt.xlabel("count")
plt.ylabel("accuracy")
plt.ylim(0, 1.0)
# グラフの表示
plt.show()
```

データ読み込み完了

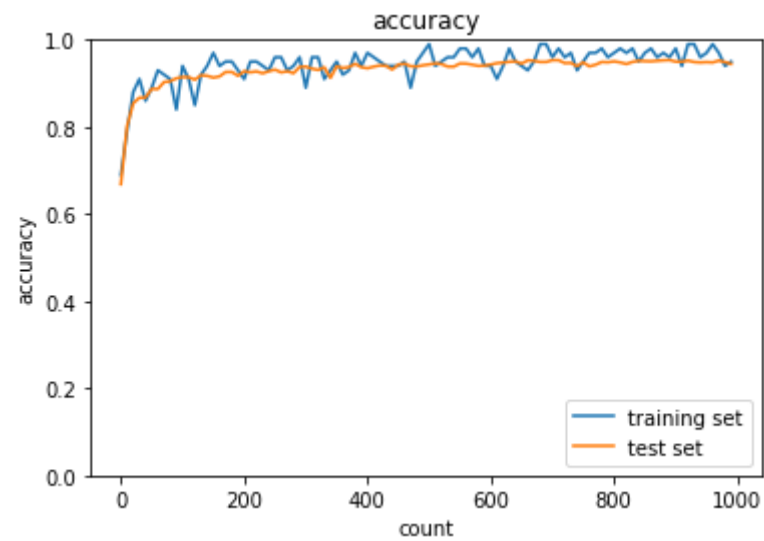
Generation: 10. 正答率(トレーニング) = 0.69
: 10. 正答率(テスト) = 0.6686
Generation: 20. 正答率(トレーニング) = 0.79
: 20. 正答率(テスト) = 0.8014
Generation: 30. 正答率(トレーニング) = 0.88
: 30. 正答率(テスト) = 0.8543
Generation: 40. 正答率(トレーニング) = 0.91
: 40. 正答率(テスト) = 0.8669
Generation: 50. 正答率(トレーニング) = 0.86
: 50. 正答率(テスト) = 0.8675
Generation: 60. 正答率(トレーニング) = 0.89
: 60. 正答率(テスト) = 0.8873
Generation: 70. 正答率(トレーニング) = 0.93
: 70. 正答率(テスト) = 0.887
Generation: 80. 正答率(トレーニング) = 0.92
: 80. 正答率(テスト) = 0.903
Generation: 90. 正答率(トレーニング) = 0.91
: 90. 正答率(テスト) = 0.9048
Generation: 100. 正答率(トレーニング) = 0.84
: 100. 正答率(テスト) = 0.9116
Generation: 110. 正答率(トレーニング) = 0.94
: 110. 正答率(テスト) = 0.9145
Generation: 120. 正答率(トレーニング) = 0.91
: 120. 正答率(テスト) = 0.9142
Generation: 130. 正答率(トレーニング) = 0.85
: 130. 正答率(テスト) = 0.9084
Generation: 140. 正答率(トレーニング) = 0.92
: 140. 正答率(テスト) = 0.9187
Generation: 150. 正答率(トレーニング) = 0.94
: 150. 正答率(テスト) = 0.9173
Generation: 160. 正答率(トレーニング) = 0.97
: 160. 正答率(テスト) = 0.9132
Generation: 170. 正答率(トレーニング) = 0.94
: 170. 正答率(テスト) = 0.9159
Generation: 180. 正答率(トレーニング) = 0.95
: 180. 正答率(テスト) = 0.9261
Generation: 190. 正答率(トレーニング) = 0.95
: 190. 正答率(テスト) = 0.9261
Generation: 200. 正答率(トレーニング) = 0.93
: 200. 正答率(テスト) = 0.9174

Generation: 210. 正答率(トレーニング) = 0.91
: 210. 正答率(テスト) = 0.9279
Generation: 220. 正答率(トレーニング) = 0.95
: 220. 正答率(テスト) = 0.9254
Generation: 230. 正答率(トレーニング) = 0.95
: 230. 正答率(テスト) = 0.9274
Generation: 240. 正答率(トレーニング) = 0.94
: 240. 正答率(テスト) = 0.9231
Generation: 250. 正答率(トレーニング) = 0.93
: 250. 正答率(テスト) = 0.9279
Generation: 260. 正答率(トレーニング) = 0.96
: 260. 正答率(テスト) = 0.9315
Generation: 270. 正答率(トレーニング) = 0.96
: 270. 正答率(テスト) = 0.9252
Generation: 280. 正答率(トレーニング) = 0.93
: 280. 正答率(テスト) = 0.9278
Generation: 290. 正答率(トレーニング) = 0.94
: 290. 正答率(テスト) = 0.9233
Generation: 300. 正答率(トレーニング) = 0.96
: 300. 正答率(テスト) = 0.9377
Generation: 310. 正答率(トレーニング) = 0.89
: 310. 正答率(テスト) = 0.9384
Generation: 320. 正答率(トレーニング) = 0.96
: 320. 正答率(テスト) = 0.9337
Generation: 330. 正答率(トレーニング) = 0.96
: 330. 正答率(テスト) = 0.9308
Generation: 340. 正答率(トレーニング) = 0.91
: 340. 正答率(テスト) = 0.9368
Generation: 350. 正答率(トレーニング) = 0.93
: 350. 正答率(テスト) = 0.9126
Generation: 360. 正答率(トレーニング) = 0.95
: 360. 正答率(テスト) = 0.9392
Generation: 370. 正答率(トレーニング) = 0.92
: 370. 正答率(テスト) = 0.9352
Generation: 380. 正答率(トレーニング) = 0.93
: 380. 正答率(テスト) = 0.9373
Generation: 390. 正答率(トレーニング) = 0.97
: 390. 正答率(テスト) = 0.9449
Generation: 400. 正答率(トレーニング) = 0.94
: 400. 正答率(テスト) = 0.9372
Generation: 410. 正答率(トレーニング) = 0.97
: 410. 正答率(テスト) = 0.9347

Generation: 420. 正答率(トレーニング) = 0.96
: 420. 正答率(テスト) = 0.9383
Generation: 430. 正答率(トレーニング) = 0.95
: 430. 正答率(テスト) = 0.9412
Generation: 440. 正答率(トレーニング) = 0.94
: 440. 正答率(テスト) = 0.9404
Generation: 450. 正答率(トレーニング) = 0.94
: 450. 正答率(テスト) = 0.9318
Generation: 460. 正答率(トレーニング) = 0.94
: 460. 正答率(テスト) = 0.9429
Generation: 470. 正答率(トレーニング) = 0.95
: 470. 正答率(テスト) = 0.9452
Generation: 480. 正答率(トレーニング) = 0.89
: 480. 正答率(テスト) = 0.9394
Generation: 490. 正答率(トレーニング) = 0.95
: 490. 正答率(テスト) = 0.9386
Generation: 500. 正答率(トレーニング) = 0.97
: 500. 正答率(テスト) = 0.9415
Generation: 510. 正答率(トレーニング) = 0.99
: 510. 正答率(テスト) = 0.9435
Generation: 520. 正答率(トレーニング) = 0.94
: 520. 正答率(テスト) = 0.9459
Generation: 530. 正答率(トレーニング) = 0.95
: 530. 正答率(テスト) = 0.9444
Generation: 540. 正答率(トレーニング) = 0.96
: 540. 正答率(テスト) = 0.9386
Generation: 550. 正答率(トレーニング) = 0.96
: 550. 正答率(テスト) = 0.9383
Generation: 560. 正答率(トレーニング) = 0.98
: 560. 正答率(テスト) = 0.9457
Generation: 570. 正答率(トレーニング) = 0.98
: 570. 正答率(テスト) = 0.9453
Generation: 580. 正答率(トレーニング) = 0.96
: 580. 正答率(テスト) = 0.9428
Generation: 590. 正答率(トレーニング) = 0.98
: 590. 正答率(テスト) = 0.9401
Generation: 600. 正答率(トレーニング) = 0.94
: 600. 正答率(テスト) = 0.9409
Generation: 610. 正答率(トレーニング) = 0.94
: 610. 正答率(テスト) = 0.9432
Generation: 620. 正答率(トレーニング) = 0.91
: 620. 正答率(テスト) = 0.9464

Generation: 630. 正答率(トレーニング) = 0.94
: 630. 正答率(テスト) = 0.9478
Generation: 640. 正答率(トレーニング) = 0.98
: 640. 正答率(テスト) = 0.9495
Generation: 650. 正答率(トレーニング) = 0.95
: 650. 正答率(テスト) = 0.9505
Generation: 660. 正答率(トレーニング) = 0.94
: 660. 正答率(テスト) = 0.9453
Generation: 670. 正答率(トレーニング) = 0.93
: 670. 正答率(テスト) = 0.9529
Generation: 680. 正答率(トレーニング) = 0.95
: 680. 正答率(テスト) = 0.9517
Generation: 690. 正答率(トレーニング) = 0.99
: 690. 正答率(テスト) = 0.9492
Generation: 700. 正答率(トレーニング) = 0.99
: 700. 正答率(テスト) = 0.9492
Generation: 710. 正答率(トレーニング) = 0.96
: 710. 正答率(テスト) = 0.9532
Generation: 720. 正答率(トレーニング) = 0.98
: 720. 正答率(テスト) = 0.9532
Generation: 730. 正答率(トレーニング) = 0.96
: 730. 正答率(テスト) = 0.9465
Generation: 740. 正答率(トレーニング) = 0.97
: 740. 正答率(テスト) = 0.9469
Generation: 750. 正答率(トレーニング) = 0.93
: 750. 正答率(テスト) = 0.9393
Generation: 760. 正答率(トレーニング) = 0.95
: 760. 正答率(テスト) = 0.9483
Generation: 770. 正答率(トレーニング) = 0.97
: 770. 正答率(テスト) = 0.939
Generation: 780. 正答率(トレーニング) = 0.97
: 780. 正答率(テスト) = 0.9428
Generation: 790. 正答率(トレーニング) = 0.98
: 790. 正答率(テスト) = 0.9491
Generation: 800. 正答率(トレーニング) = 0.96
: 800. 正答率(テスト) = 0.9484
Generation: 810. 正答率(トレーニング) = 0.97
: 810. 正答率(テスト) = 0.9505
Generation: 820. 正答率(トレーニング) = 0.98
: 820. 正答率(テスト) = 0.9481
Generation: 830. 正答率(トレーニング) = 0.97
: 830. 正答率(テスト) = 0.9445

Generation: 840. 正答率(トレーニング) = 0.98
: 840. 正答率(テスト) = 0.9493
Generation: 850. 正答率(トレーニング) = 0.95
: 850. 正答率(テスト) = 0.9519
Generation: 860. 正答率(トレーニング) = 0.97
: 860. 正答率(テスト) = 0.9511
Generation: 870. 正答率(トレーニング) = 0.98
: 870. 正答率(テスト) = 0.9507
Generation: 880. 正答率(トレーニング) = 0.96
: 880. 正答率(テスト) = 0.9518
Generation: 890. 正答率(トレーニング) = 0.97
: 890. 正答率(テスト) = 0.9529
Generation: 900. 正答率(トレーニング) = 0.96
: 900. 正答率(テスト) = 0.9535
Generation: 910. 正答率(トレーニング) = 0.98
: 910. 正答率(テスト) = 0.9499
Generation: 920. 正答率(トレーニング) = 0.94
: 920. 正答率(テスト) = 0.9497
Generation: 930. 正答率(トレーニング) = 0.99
: 930. 正答率(テスト) = 0.9522
Generation: 940. 正答率(トレーニング) = 0.99
: 940. 正答率(テスト) = 0.949
Generation: 950. 正答率(トレーニング) = 0.96
: 950. 正答率(テスト) = 0.948
Generation: 960. 正答率(トレーニング) = 0.97
: 960. 正答率(テスト) = 0.9488
Generation: 970. 正答率(トレーニング) = 0.99
: 970. 正答率(テスト) = 0.948
Generation: 980. 正答率(トレーニング) = 0.97
: 980. 正答率(テスト) = 0.9523
Generation: 990. 正答率(トレーニング) = 0.94
: 990. 正答率(テスト) = 0.9488
Generation: 1000. 正答率(トレーニング) = 0.95
: 1000. 正答率(テスト) = 0.9446



[try] バッチ正規化をしてみよう & ReLU, He

Adam

```

In [52]: # データの読み込み
(x_train, d_train), (x_test, d_test) = load_mnist(normalize=True, one_hot_label=True)

print("データ読み込み完了")

# batch_normalizationの設定 =====
use_batchnorm = True
#use_batchnorm = False
# =====

network = MultiLayerNet(input_size=784, hidden_size_list=[40, 20], output_size=10, activation='relu', weight_init_std='he',
                        use_batchnorm=use_batchnorm)

iters_num = 1000
train_size = x_train.shape[0]
batch_size = 100
learning_rate = 0.01
beta1 = 0.9
beta2 = 0.999

train_loss_list = []
accuracies_train = []
accuracies_test = []

plot_interval=10

for i in range(iters_num):
    batch_mask = np.random.choice(train_size, batch_size)
    x_batch = x_train[batch_mask]
    d_batch = d_train[batch_mask]

    # 勾配
    grad = network.gradient(x_batch, d_batch)
    if i == 0:
        m = {}
        v = {}
    learning_rate_t = learning_rate * np.sqrt(1.0 - beta2 ** (i + 1)) / (1.0 - beta1 ** (i + 1))
    for key in ('W1', 'W2', 'W3', 'b1', 'b2', 'b3'):
        if i == 0:
            m[key] = np.zeros_like(network.params[key])
            v[key] = np.zeros_like(network.params[key])

```

```
m[key] += (1 - beta1) * (grad[key] - m[key])
v[key] += (1 - beta2) * (grad[key] ** 2 - v[key])
network.params[key] -= learning_rate_t * m[key] / (np.sqrt(v[key]) + 1e-7)

if (i + 1) % plot_interval == 0:
    accr_test = network.accuracy(x_test, d_test)
    accuracies_test.append(accr_test)
    accr_train = network.accuracy(x_batch, d_batch)
    accuracies_train.append(accr_train)
    loss = network.loss(x_batch, d_batch)
    train_loss_list.append(loss)

    print('Generation: ' + str(i+1) + '. 正答率(トレーニング) = ' + str(accr_train))
    print('          : ' + str(i+1) + '. 正答率(テスト) = ' + str(accr_test))

lists = range(0, iters_num, plot_interval)
plt.plot(lists, accuracies_train, label="training set")
plt.plot(lists, accuracies_test, label="test set")
plt.legend(loc="lower right")
plt.title("accuracy")
plt.xlabel("count")
plt.ylabel("accuracy")
plt.ylim(0, 1.0)
# グラフの表示
plt.show()
```

データ読み込み完了

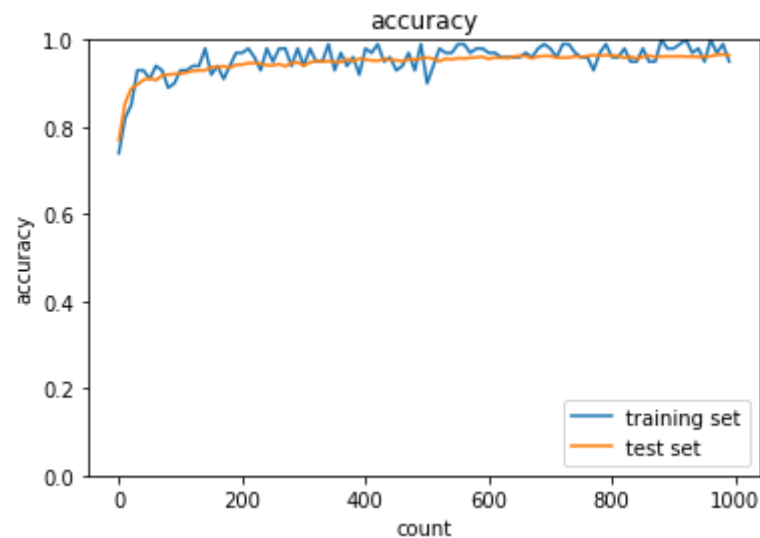
Generation: 10. 正答率(トレーニング) = 0.74
: 10. 正答率(テスト) = 0.7696
Generation: 20. 正答率(トレーニング) = 0.82
: 20. 正答率(テスト) = 0.8543
Generation: 30. 正答率(トレーニング) = 0.85
: 30. 正答率(テスト) = 0.8882
Generation: 40. 正答率(トレーニング) = 0.93
: 40. 正答率(テスト) = 0.8971
Generation: 50. 正答率(トレーニング) = 0.93
: 50. 正答率(テスト) = 0.908
Generation: 60. 正答率(トレーニング) = 0.91
: 60. 正答率(テスト) = 0.9123
Generation: 70. 正答率(トレーニング) = 0.94
: 70. 正答率(テスト) = 0.9069
Generation: 80. 正答率(トレーニング) = 0.93
: 80. 正答率(テスト) = 0.9178
Generation: 90. 正答率(トレーニング) = 0.89
: 90. 正答率(テスト) = 0.9201
Generation: 100. 正答率(トレーニング) = 0.9
: 100. 正答率(テスト) = 0.9216
Generation: 110. 正答率(トレーニング) = 0.93
: 110. 正答率(テスト) = 0.9224
Generation: 120. 正答率(トレーニング) = 0.93
: 120. 正答率(テスト) = 0.9248
Generation: 130. 正答率(トレーニング) = 0.94
: 130. 正答率(テスト) = 0.9287
Generation: 140. 正答率(トレーニング) = 0.94
: 140. 正答率(テスト) = 0.93
Generation: 150. 正答率(トレーニング) = 0.98
: 150. 正答率(テスト) = 0.9297
Generation: 160. 正答率(トレーニング) = 0.92
: 160. 正答率(テスト) = 0.9373
Generation: 170. 正答率(トレーニング) = 0.94
: 170. 正答率(テスト) = 0.9369
Generation: 180. 正答率(トレーニング) = 0.91
: 180. 正答率(テスト) = 0.9393
Generation: 190. 正答率(トレーニング) = 0.94
: 190. 正答率(テスト) = 0.936
Generation: 200. 正答率(トレーニング) = 0.97
: 200. 正答率(テスト) = 0.9426

Generation: 210. 正答率(トレーニング) = 0.97
: 210. 正答率(テスト) = 0.943
Generation: 220. 正答率(トレーニング) = 0.98
: 220. 正答率(テスト) = 0.9465
Generation: 230. 正答率(トレーニング) = 0.96
: 230. 正答率(テスト) = 0.9468
Generation: 240. 正答率(トレーニング) = 0.93
: 240. 正答率(テスト) = 0.9449
Generation: 250. 正答率(トレーニング) = 0.98
: 250. 正答率(テスト) = 0.9418
Generation: 260. 正答率(トレーニング) = 0.95
: 260. 正答率(テスト) = 0.9411
Generation: 270. 正答率(トレーニング) = 0.98
: 270. 正答率(テスト) = 0.9445
Generation: 280. 正答率(トレーニング) = 0.98
: 280. 正答率(テスト) = 0.9388
Generation: 290. 正答率(トレーニング) = 0.94
: 290. 正答率(テスト) = 0.9476
Generation: 300. 正答率(トレーニング) = 0.98
: 300. 正答率(テスト) = 0.9482
Generation: 310. 正答率(トレーニング) = 0.94
: 310. 正答率(テスト) = 0.9409
Generation: 320. 正答率(トレーニング) = 0.98
: 320. 正答率(テスト) = 0.948
Generation: 330. 正答率(トレーニング) = 0.95
: 330. 正答率(テスト) = 0.9518
Generation: 340. 正答率(トレーニング) = 0.95
: 340. 正答率(テスト) = 0.9498
Generation: 350. 正答率(トレーニング) = 0.99
: 350. 正答率(テスト) = 0.951
Generation: 360. 正答率(トレーニング) = 0.93
: 360. 正答率(テスト) = 0.9491
Generation: 370. 正答率(トレーニング) = 0.97
: 370. 正答率(テスト) = 0.9487
Generation: 380. 正答率(トレーニング) = 0.94
: 380. 正答率(テスト) = 0.9556
Generation: 390. 正答率(トレーニング) = 0.96
: 390. 正答率(テスト) = 0.9512
Generation: 400. 正答率(トレーニング) = 0.92
: 400. 正答率(テスト) = 0.9569
Generation: 410. 正答率(トレーニング) = 0.98
: 410. 正答率(テスト) = 0.9548

Generation: 420. 正答率(トレーニング) = 0.97
: 420. 正答率(テスト) = 0.9526
Generation: 430. 正答率(トレーニング) = 0.99
: 430. 正答率(テスト) = 0.9519
Generation: 440. 正答率(トレーニング) = 0.95
: 440. 正答率(テスト) = 0.9563
Generation: 450. 正答率(トレーニング) = 0.96
: 450. 正答率(テスト) = 0.9534
Generation: 460. 正答率(トレーニング) = 0.93
: 460. 正答率(テスト) = 0.9535
Generation: 470. 正答率(トレーニング) = 0.94
: 470. 正答率(テスト) = 0.9507
Generation: 480. 正答率(トレーニング) = 0.97
: 480. 正答率(テスト) = 0.9557
Generation: 490. 正答率(トレーニング) = 0.93
: 490. 正答率(テスト) = 0.9542
Generation: 500. 正答率(トレーニング) = 0.99
: 500. 正答率(テスト) = 0.9574
Generation: 510. 正答率(トレーニング) = 0.9
: 510. 正答率(テスト) = 0.959
Generation: 520. 正答率(トレーニング) = 0.94
: 520. 正答率(テスト) = 0.9567
Generation: 530. 正答率(トレーニング) = 0.98
: 530. 正答率(テスト) = 0.9509
Generation: 540. 正答率(トレーニング) = 0.97
: 540. 正答率(テスト) = 0.9567
Generation: 550. 正答率(トレーニング) = 0.97
: 550. 正答率(テスト) = 0.9549
Generation: 560. 正答率(トレーニング) = 0.99
: 560. 正答率(テスト) = 0.9579
Generation: 570. 正答率(トレーニング) = 0.99
: 570. 正答率(テスト) = 0.9572
Generation: 580. 正答率(トレーニング) = 0.97
: 580. 正答率(テスト) = 0.9588
Generation: 590. 正答率(トレーニング) = 0.98
: 590. 正答率(テスト) = 0.9597
Generation: 600. 正答率(トレーニング) = 0.98
: 600. 正答率(テスト) = 0.9608
Generation: 610. 正答率(トレーニング) = 0.97
: 610. 正答率(テスト) = 0.9563
Generation: 620. 正答率(トレーニング) = 0.97
: 620. 正答率(テスト) = 0.9591

Generation: 630. 正答率(トレーニング) = 0.96
: 630. 正答率(テスト) = 0.9589
Generation: 640. 正答率(トレーニング) = 0.96
: 640. 正答率(テスト) = 0.9575
Generation: 650. 正答率(トレーニング) = 0.96
: 650. 正答率(テスト) = 0.9618
Generation: 660. 正答率(トレーニング) = 0.96
: 660. 正答率(テスト) = 0.963
Generation: 670. 正答率(トレーニング) = 0.97
: 670. 正答率(テスト) = 0.9626
Generation: 680. 正答率(トレーニング) = 0.96
: 680. 正答率(テスト) = 0.9573
Generation: 690. 正答率(トレーニング) = 0.98
: 690. 正答率(テスト) = 0.9617
Generation: 700. 正答率(トレーニング) = 0.99
: 700. 正答率(テスト) = 0.9629
Generation: 710. 正答率(トレーニング) = 0.98
: 710. 正答率(テスト) = 0.9629
Generation: 720. 正答率(トレーニング) = 0.96
: 720. 正答率(テスト) = 0.9591
Generation: 730. 正答率(トレーニング) = 0.99
: 730. 正答率(テスト) = 0.9591
Generation: 740. 正答率(トレーニング) = 0.99
: 740. 正答率(テスト) = 0.959
Generation: 750. 正答率(トレーニング) = 0.97
: 750. 正答率(テスト) = 0.9622
Generation: 760. 正答率(トレーニング) = 0.96
: 760. 正答率(テスト) = 0.9608
Generation: 770. 正答率(トレーニング) = 0.96
: 770. 正答率(テスト) = 0.9629
Generation: 780. 正答率(トレーニング) = 0.93
: 780. 正答率(テスト) = 0.9646
Generation: 790. 正答率(トレーニング) = 0.97
: 790. 正答率(テスト) = 0.9632
Generation: 800. 正答率(トレーニング) = 0.99
: 800. 正答率(テスト) = 0.965
Generation: 810. 正答率(トレーニング) = 0.96
: 810. 正答率(テスト) = 0.9634
Generation: 820. 正答率(トレーニング) = 0.96
: 820. 正答率(テスト) = 0.9633
Generation: 830. 正答率(トレーニング) = 0.98
: 830. 正答率(テスト) = 0.9584

Generation: 840. 正答率(トレーニング) = 0.95
: 840. 正答率(テスト) = 0.9611
Generation: 850. 正答率(トレーニング) = 0.95
: 850. 正答率(テスト) = 0.957
Generation: 860. 正答率(トレーニング) = 0.98
: 860. 正答率(テスト) = 0.9607
Generation: 870. 正答率(トレーニング) = 0.95
: 870. 正答率(テスト) = 0.9638
Generation: 880. 正答率(トレーニング) = 0.95
: 880. 正答率(テスト) = 0.9607
Generation: 890. 正答率(トレーニング) = 1.0
: 890. 正答率(テスト) = 0.9616
Generation: 900. 正答率(トレーニング) = 0.98
: 900. 正答率(テスト) = 0.9622
Generation: 910. 正答率(トレーニング) = 0.98
: 910. 正答率(テスト) = 0.9619
Generation: 920. 正答率(トレーニング) = 0.99
: 920. 正答率(テスト) = 0.9622
Generation: 930. 正答率(トレーニング) = 1.0
: 930. 正答率(テスト) = 0.9615
Generation: 940. 正答率(トレーニング) = 0.97
: 940. 正答率(テスト) = 0.9616
Generation: 950. 正答率(トレーニング) = 0.98
: 950. 正答率(テスト) = 0.9607
Generation: 960. 正答率(トレーニング) = 0.95
: 960. 正答率(テスト) = 0.9613
Generation: 970. 正答率(トレーニング) = 1.0
: 970. 正答率(テスト) = 0.9624
Generation: 980. 正答率(トレーニング) = 0.97
: 980. 正答率(テスト) = 0.9653
Generation: 990. 正答率(トレーニング) = 0.99
: 990. 正答率(テスト) = 0.9663
Generation: 1000. 正答率(トレーニング) = 0.95
: 1000. 正答率(テスト) = 0.9643



[try] 学習率を変えてみよう

[try] 活性化関数と重みの初期化方法を変えてみよう

初期状態ではsigmoid - gauss

activationはReLU、weight_init_stdは別の数値や'Xavier'・'He'に変更可能

[try] バッチ正規化を試みよう

use_batchnormをTrueにしよう
