

overfitting

```
In [2]: 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
from common import optimizer

(x_train, d_train), (x_test, d_test) = load_mnist(normalize=True)

print("データ読み込み完了")

# 過学習を再現するために、学習データを削減
x_train = x_train[:300]
d_train = d_train[:300]

network = MultiLayerNet(input_size=784, hidden_size_list=[100, 100, 100, 100, 100, 100], output_size=10)
optimizer = optimizer.SGD(learning_rate=0.01)

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

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]

    grad = network.gradient(x_batch, d_batch)
    optimizer.update(network.params, grad)
```

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

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

    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

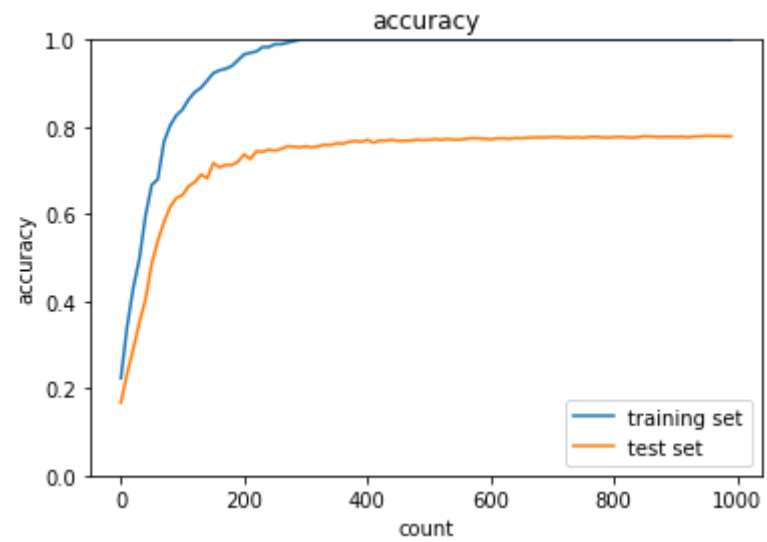
Generation: 10. 正答率(トレーニング) = 0.2233333333333333
: 10. 正答率(テスト) = 0.1672
Generation: 20. 正答率(トレーニング) = 0.34
: 20. 正答率(テスト) = 0.2329
Generation: 30. 正答率(トレーニング) = 0.43
: 30. 正答率(テスト) = 0.2895
Generation: 40. 正答率(トレーニング) = 0.49666666666666665
: 40. 正答率(テスト) = 0.3507
Generation: 50. 正答率(トレーニング) = 0.5966666666666667
: 50. 正答率(テスト) = 0.4038
Generation: 60. 正答率(トレーニング) = 0.6666666666666666
: 60. 正答率(テスト) = 0.4862
Generation: 70. 正答率(トレーニング) = 0.68
: 70. 正答率(テスト) = 0.5385
Generation: 80. 正答率(トレーニング) = 0.7666666666666667
: 80. 正答率(テスト) = 0.5819
Generation: 90. 正答率(トレーニング) = 0.8033333333333333
: 90. 正答率(テスト) = 0.6168
Generation: 100. 正答率(トレーニング) = 0.8266666666666667
: 100. 正答率(テスト) = 0.6368
Generation: 110. 正答率(トレーニング) = 0.84
: 110. 正答率(テスト) = 0.6439
Generation: 120. 正答率(トレーニング) = 0.8633333333333333
: 120. 正答率(テスト) = 0.6639
Generation: 130. 正答率(トレーニング) = 0.88
: 130. 正答率(テスト) = 0.6737
Generation: 140. 正答率(トレーニング) = 0.89
: 140. 正答率(テスト) = 0.6905
Generation: 150. 正答率(トレーニング) = 0.9066666666666666
: 150. 正答率(テスト) = 0.6821
Generation: 160. 正答率(トレーニング) = 0.9233333333333333
: 160. 正答率(テスト) = 0.7171
Generation: 170. 正答率(トレーニング) = 0.93
: 170. 正答率(テスト) = 0.7072
Generation: 180. 正答率(トレーニング) = 0.9333333333333333
: 180. 正答率(テスト) = 0.7127
Generation: 190. 正答率(トレーニング) = 0.94
: 190. 正答率(テスト) = 0.7124
Generation: 200. 正答率(トレーニング) = 0.9533333333333334
: 200. 正答率(テスト) = 0.7202

Generation: 210. 正答率(トレーニング) = 0.9666666666666667
: 210. 正答率(テスト) = 0.7368
Generation: 220. 正答率(トレーニング) = 0.97
: 220. 正答率(テスト) = 0.7264
Generation: 230. 正答率(トレーニング) = 0.9733333333333334
: 230. 正答率(テスト) = 0.7444
Generation: 240. 正答率(トレーニング) = 0.9833333333333333
: 240. 正答率(テスト) = 0.7429
Generation: 250. 正答率(トレーニング) = 0.9833333333333333
: 250. 正答率(テスト) = 0.7482
Generation: 260. 正答率(トレーニング) = 0.99
: 260. 正答率(テスト) = 0.7455
Generation: 270. 正答率(トレーニング) = 0.99
: 270. 正答率(テスト) = 0.7493
Generation: 280. 正答率(トレーニング) = 0.9933333333333333
: 280. 正答率(テスト) = 0.7557
Generation: 290. 正答率(トレーニング) = 0.9966666666666667
: 290. 正答率(テスト) = 0.7546
Generation: 300. 正答率(トレーニング) = 1.0
: 300. 正答率(テスト) = 0.7531
Generation: 310. 正答率(トレーニング) = 1.0
: 310. 正答率(テスト) = 0.7555
Generation: 320. 正答率(トレーニング) = 1.0
: 320. 正答率(テスト) = 0.7532
Generation: 330. 正答率(トレーニング) = 1.0
: 330. 正答率(テスト) = 0.7558
Generation: 340. 正答率(トレーニング) = 1.0
: 340. 正答率(テスト) = 0.759
Generation: 350. 正答率(トレーニング) = 1.0
: 350. 正答率(テスト) = 0.7581
Generation: 360. 正答率(トレーニング) = 1.0
: 360. 正答率(テスト) = 0.7626
Generation: 370. 正答率(トレーニング) = 1.0
: 370. 正答率(テスト) = 0.7614
Generation: 380. 正答率(トレーニング) = 1.0
: 380. 正答率(テスト) = 0.7659
Generation: 390. 正答率(トレーニング) = 1.0
: 390. 正答率(テスト) = 0.7677
Generation: 400. 正答率(トレーニング) = 1.0
: 400. 正答率(テスト) = 0.7659
Generation: 410. 正答率(トレーニング) = 1.0
: 410. 正答率(テスト) = 0.7703

Generation: 420. 正答率(トレーニング) = 1.0
: 420. 正答率(テスト) = 0.7638
Generation: 430. 正答率(トレーニング) = 1.0
: 430. 正答率(テスト) = 0.7689
Generation: 440. 正答率(トレーニング) = 1.0
: 440. 正答率(テスト) = 0.7686
Generation: 450. 正答率(トレーニング) = 1.0
: 450. 正答率(テスト) = 0.7705
Generation: 460. 正答率(トレーニング) = 1.0
: 460. 正答率(テスト) = 0.7676
Generation: 470. 正答率(トレーニング) = 1.0
: 470. 正答率(テスト) = 0.768
Generation: 480. 正答率(トレーニング) = 1.0
: 480. 正答率(テスト) = 0.7686
Generation: 490. 正答率(トレーニング) = 1.0
: 490. 正答率(テスト) = 0.7708
Generation: 500. 正答率(トレーニング) = 1.0
: 500. 正答率(テスト) = 0.7694
Generation: 510. 正答率(トレーニング) = 1.0
: 510. 正答率(テスト) = 0.7705
Generation: 520. 正答率(トレーニング) = 1.0
: 520. 正答率(テスト) = 0.7718
Generation: 530. 正答率(トレーニング) = 1.0
: 530. 正答率(テスト) = 0.7704
Generation: 540. 正答率(トレーニング) = 1.0
: 540. 正答率(テスト) = 0.7722
Generation: 550. 正答率(トレーニング) = 1.0
: 550. 正答率(テスト) = 0.7707
Generation: 560. 正答率(トレーニング) = 1.0
: 560. 正答率(テスト) = 0.7708
Generation: 570. 正答率(トレーニング) = 1.0
: 570. 正答率(テスト) = 0.7726
Generation: 580. 正答率(トレーニング) = 1.0
: 580. 正答率(テスト) = 0.7742
Generation: 590. 正答率(トレーニング) = 1.0
: 590. 正答率(テスト) = 0.7738
Generation: 600. 正答率(トレーニング) = 1.0
: 600. 正答率(テスト) = 0.7732
Generation: 610. 正答率(トレーニング) = 1.0
: 610. 正答率(テスト) = 0.7714
Generation: 620. 正答率(トレーニング) = 1.0
: 620. 正答率(テスト) = 0.7736

Generation: 630. 正答率(トレーニング) = 1.0
: 630. 正答率(テスト) = 0.7738
Generation: 640. 正答率(トレーニング) = 1.0
: 640. 正答率(テスト) = 0.7727
Generation: 650. 正答率(トレーニング) = 1.0
: 650. 正答率(テスト) = 0.7747
Generation: 660. 正答率(トレーニング) = 1.0
: 660. 正答率(テスト) = 0.7741
Generation: 670. 正答率(トレーニング) = 1.0
: 670. 正答率(テスト) = 0.7753
Generation: 680. 正答率(トレーニング) = 1.0
: 680. 正答率(テスト) = 0.7758
Generation: 690. 正答率(トレーニング) = 1.0
: 690. 正答率(テスト) = 0.7763
Generation: 700. 正答率(トレーニング) = 1.0
: 700. 正答率(テスト) = 0.7759
Generation: 710. 正答率(トレーニング) = 1.0
: 710. 正答率(テスト) = 0.7769
Generation: 720. 正答率(トレーニング) = 1.0
: 720. 正答率(テスト) = 0.777
Generation: 730. 正答率(トレーニング) = 1.0
: 730. 正答率(テスト) = 0.776
Generation: 740. 正答率(トレーニング) = 1.0
: 740. 正答率(テスト) = 0.7754
Generation: 750. 正答率(トレーニング) = 1.0
: 750. 正答率(テスト) = 0.7765
Generation: 760. 正答率(トレーニング) = 1.0
: 760. 正答率(テスト) = 0.7752
Generation: 770. 正答率(トレーニング) = 1.0
: 770. 正答率(テスト) = 0.7771
Generation: 780. 正答率(トレーニング) = 1.0
: 780. 正答率(テスト) = 0.7775
Generation: 790. 正答率(トレーニング) = 1.0
: 790. 正答率(テスト) = 0.7761
Generation: 800. 正答率(トレーニング) = 1.0
: 800. 正答率(テスト) = 0.7755
Generation: 810. 正答率(トレーニング) = 1.0
: 810. 正答率(テスト) = 0.7766
Generation: 820. 正答率(トレーニング) = 1.0
: 820. 正答率(テスト) = 0.7773
Generation: 830. 正答率(トレーニング) = 1.0
: 830. 正答率(テスト) = 0.7763

Generation: 840. 正答率(トレーニング) = 1.0
: 840. 正答率(テスト) = 0.7757
Generation: 850. 正答率(トレーニング) = 1.0
: 850. 正答率(テスト) = 0.7769
Generation: 860. 正答率(トレーニング) = 1.0
: 860. 正答率(テスト) = 0.7785
Generation: 870. 正答率(トレーニング) = 1.0
: 870. 正答率(テスト) = 0.7778
Generation: 880. 正答率(トレーニング) = 1.0
: 880. 正答率(テスト) = 0.777
Generation: 890. 正答率(トレーニング) = 1.0
: 890. 正答率(テスト) = 0.7772
Generation: 900. 正答率(トレーニング) = 1.0
: 900. 正答率(テスト) = 0.7777
Generation: 910. 正答率(トレーニング) = 1.0
: 910. 正答率(テスト) = 0.7769
Generation: 920. 正答率(トレーニング) = 1.0
: 920. 正答率(テスト) = 0.7779
Generation: 930. 正答率(トレーニング) = 1.0
: 930. 正答率(テスト) = 0.7764
Generation: 940. 正答率(トレーニング) = 1.0
: 940. 正答率(テスト) = 0.7779
Generation: 950. 正答率(トレーニング) = 1.0
: 950. 正答率(テスト) = 0.7786
Generation: 960. 正答率(トレーニング) = 1.0
: 960. 正答率(テスト) = 0.7794
Generation: 970. 正答率(トレーニング) = 1.0
: 970. 正答率(テスト) = 0.7791
Generation: 980. 正答率(トレーニング) = 1.0
: 980. 正答率(テスト) = 0.779
Generation: 990. 正答率(トレーニング) = 1.0
: 990. 正答率(テスト) = 0.7785
Generation: 1000. 正答率(トレーニング) = 1.0
: 1000. 正答率(テスト) = 0.7784



weight decay

L2

```
In [3]: from common import optimizer

(x_train, d_train), (x_test, d_test) = load_mnist(normalize=True)

print("データ読み込み完了")

# 過学習を再現するために、学習データを削減
x_train = x_train[:300]
d_train = d_train[:300]

network = MultiLayerNet(input_size=784, hidden_size_list=[100, 100, 100, 100, 100, 100], output_size=10)

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
hidden_layer_num = network.hidden_layer_num

# 正則化強度設定 =====
weight_decay_lambda = 0.1
# =====

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]

    grad = network.gradient(x_batch, d_batch)
    weight_decay = 0

    for idx in range(1, hidden_layer_num+1):
        grad['W' + str(idx)] = network.layers['Affine' + str(idx)].dW + weight_decay_lambda * network.params['W' + str(idx)]
        grad['b' + str(idx)] = network.layers['Affine' + str(idx)].db
```

```
network.params['W' + str(idx)] -= learning_rate * grad['W' + str(idx)]
network.params['b' + str(idx)] -= learning_rate * grad['b' + str(idx)]
weight_decay += 0.5 * weight_decay_lambda * np.sqrt(np.sum(network.params['W' + str(idx)] ** 2))

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

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

    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

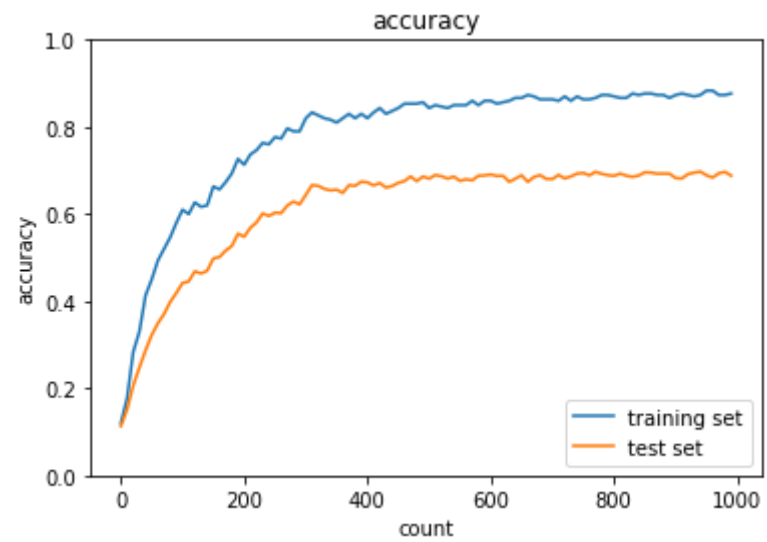
Generation: 10. 正答率(トレーニング) = 0.12
: 10. 正答率(テスト) = 0.1133
Generation: 20. 正答率(トレーニング) = 0.17666666666666667
: 20. 正答率(テスト) = 0.151
Generation: 30. 正答率(トレーニング) = 0.2833333333333333
: 30. 正答率(テスト) = 0.206
Generation: 40. 正答率(トレーニング) = 0.33
: 40. 正答率(テスト) = 0.2475
Generation: 50. 正答率(トレーニング) = 0.4133333333333333
: 50. 正答率(テスト) = 0.2872
Generation: 60. 正答率(トレーニング) = 0.45
: 60. 正答率(テスト) = 0.3224
Generation: 70. 正答率(トレーニング) = 0.4933333333333333
: 70. 正答率(テスト) = 0.3491
Generation: 80. 正答率(トレーニング) = 0.52
: 80. 正答率(テスト) = 0.3706
Generation: 90. 正答率(トレーニング) = 0.5466666666666666
: 90. 正答率(テスト) = 0.3983
Generation: 100. 正答率(トレーニング) = 0.58
: 100. 正答率(テスト) = 0.419
Generation: 110. 正答率(トレーニング) = 0.61
: 110. 正答率(テスト) = 0.4418
Generation: 120. 正答率(トレーニング) = 0.6
: 120. 正答率(テスト) = 0.4453
Generation: 130. 正答率(トレーニング) = 0.62666666666666667
: 130. 正答率(テスト) = 0.4685
Generation: 140. 正答率(トレーニング) = 0.61666666666666667
: 140. 正答率(テスト) = 0.4637
Generation: 150. 正答率(トレーニング) = 0.62
: 150. 正答率(テスト) = 0.4697
Generation: 160. 正答率(トレーニング) = 0.6633333333333333
: 160. 正答率(テスト) = 0.498
Generation: 170. 正答率(トレーニング) = 0.6566666666666666
: 170. 正答率(テスト) = 0.5015
Generation: 180. 正答率(トレーニング) = 0.6733333333333333
: 180. 正答率(テスト) = 0.5162
Generation: 190. 正答率(トレーニング) = 0.6933333333333334
: 190. 正答率(テスト) = 0.5267
Generation: 200. 正答率(トレーニング) = 0.72666666666666667
: 200. 正答率(テスト) = 0.5549

Generation: 210. 正答率(トレーニング) = 0.7133333333333334
: 210. 正答率(テスト) = 0.548
Generation: 220. 正答率(トレーニング) = 0.7366666666666667
: 220. 正答率(テスト) = 0.5685
Generation: 230. 正答率(トレーニング) = 0.7466666666666667
: 230. 正答率(テスト) = 0.5808
Generation: 240. 正答率(トレーニング) = 0.7633333333333333
: 240. 正答率(テスト) = 0.6014
Generation: 250. 正答率(トレーニング) = 0.76
: 250. 正答率(テスト) = 0.5956
Generation: 260. 正答率(トレーニング) = 0.7766666666666666
: 260. 正答率(テスト) = 0.6031
Generation: 270. 正答率(トレーニング) = 0.7733333333333333
: 270. 正答率(テスト) = 0.6021
Generation: 280. 正答率(トレーニング) = 0.7966666666666666
: 280. 正答率(テスト) = 0.6196
Generation: 290. 正答率(トレーニング) = 0.79
: 290. 正答率(テスト) = 0.6287
Generation: 300. 正答率(トレーニング) = 0.79
: 300. 正答率(テスト) = 0.6223
Generation: 310. 正答率(トレーニング) = 0.82
: 310. 正答率(テスト) = 0.6441
Generation: 320. 正答率(トレーニング) = 0.8333333333333334
: 320. 正答率(テスト) = 0.6665
Generation: 330. 正答率(トレーニング) = 0.8266666666666667
: 330. 正答率(テスト) = 0.6648
Generation: 340. 正答率(トレーニング) = 0.82
: 340. 正答率(テスト) = 0.6579
Generation: 350. 正答率(トレーニング) = 0.8166666666666667
: 350. 正答率(テスト) = 0.6545
Generation: 360. 正答率(トレーニング) = 0.81
: 360. 正答率(テスト) = 0.656
Generation: 370. 正答率(トレーニング) = 0.82
: 370. 正答率(テスト) = 0.6486
Generation: 380. 正答率(トレーニング) = 0.83
: 380. 正答率(テスト) = 0.6664
Generation: 390. 正答率(トレーニング) = 0.82
: 390. 正答率(テスト) = 0.6648
Generation: 400. 正答率(トレーニング) = 0.83
: 400. 正答率(テスト) = 0.6743
Generation: 410. 正答率(トレーニング) = 0.82
: 410. 正答率(テスト) = 0.6728

Generation: 420. 正答率(トレーニング) = 0.8333333333333334
: 420. 正答率(テスト) = 0.6657
Generation: 430. 正答率(トレーニング) = 0.8433333333333334
: 430. 正答率(テスト) = 0.6713
Generation: 440. 正答率(トレーニング) = 0.83
: 440. 正答率(テスト) = 0.6608
Generation: 450. 正答率(トレーニング) = 0.8366666666666667
: 450. 正答率(テスト) = 0.6643
Generation: 460. 正答率(トレーニング) = 0.8433333333333334
: 460. 正答率(テスト) = 0.6717
Generation: 470. 正答率(トレーニング) = 0.8533333333333334
: 470. 正答率(テスト) = 0.6757
Generation: 480. 正答率(トレーニング) = 0.8533333333333334
: 480. 正答率(テスト) = 0.6864
Generation: 490. 正答率(トレーニング) = 0.8533333333333334
: 490. 正答率(テスト) = 0.6755
Generation: 500. 正答率(トレーニング) = 0.8566666666666667
: 500. 正答率(テスト) = 0.6863
Generation: 510. 正答率(トレーニング) = 0.8433333333333334
: 510. 正答率(テスト) = 0.6812
Generation: 520. 正答率(トレーニング) = 0.85
: 520. 正答率(テスト) = 0.6897
Generation: 530. 正答率(トレーニング) = 0.8466666666666667
: 530. 正答率(テスト) = 0.6868
Generation: 540. 正答率(トレーニング) = 0.8433333333333334
: 540. 正答率(テスト) = 0.6814
Generation: 550. 正答率(トレーニング) = 0.85
: 550. 正答率(テスト) = 0.6861
Generation: 560. 正答率(トレーニング) = 0.85
: 560. 正答率(テスト) = 0.6762
Generation: 570. 正答率(トレーニング) = 0.85
: 570. 正答率(テスト) = 0.6798
Generation: 580. 正答率(トレーニング) = 0.86
: 580. 正答率(テスト) = 0.6775
Generation: 590. 正答率(トレーニング) = 0.85
: 590. 正答率(テスト) = 0.688
Generation: 600. 正答率(トレーニング) = 0.86
: 600. 正答率(テスト) = 0.6887
Generation: 610. 正答率(トレーニング) = 0.86
: 610. 正答率(テスト) = 0.6908
Generation: 620. 正答率(トレーニング) = 0.8533333333333334
: 620. 正答率(テスト) = 0.6875

Generation: 630. 正答率(トレーニング) = 0.8566666666666667
: 630. 正答率(テスト) = 0.6878
Generation: 640. 正答率(トレーニング) = 0.86
: 640. 正答率(テスト) = 0.6741
Generation: 650. 正答率(トレーニング) = 0.8666666666666667
: 650. 正答率(テスト) = 0.6813
Generation: 660. 正答率(トレーニング) = 0.8666666666666667
: 660. 正答率(テスト) = 0.689
Generation: 670. 正答率(トレーニング) = 0.8733333333333333
: 670. 正答率(テスト) = 0.6739
Generation: 680. 正答率(トレーニング) = 0.87
: 680. 正答率(テスト) = 0.685
Generation: 690. 正答率(トレーニング) = 0.8633333333333333
: 690. 正答率(テスト) = 0.6895
Generation: 700. 正答率(トレーニング) = 0.8633333333333333
: 700. 正答率(テスト) = 0.6808
Generation: 710. 正答率(トレーニング) = 0.8633333333333333
: 710. 正答率(テスト) = 0.6804
Generation: 720. 正答率(トレーニング) = 0.86
: 720. 正答率(テスト) = 0.6898
Generation: 730. 正答率(トレーニング) = 0.87
: 730. 正答率(テスト) = 0.6818
Generation: 740. 正答率(トレーニング) = 0.86
: 740. 正答率(テスト) = 0.6865
Generation: 750. 正答率(トレーニング) = 0.87
: 750. 正答率(テスト) = 0.6927
Generation: 760. 正答率(トレーニング) = 0.8633333333333333
: 760. 正答率(テスト) = 0.6943
Generation: 770. 正答率(トレーニング) = 0.8633333333333333
: 770. 正答率(テスト) = 0.6889
Generation: 780. 正答率(トレーニング) = 0.8666666666666667
: 780. 正答率(テスト) = 0.6969
Generation: 790. 正答率(トレーニング) = 0.8733333333333333
: 790. 正答率(テスト) = 0.6924
Generation: 800. 正答率(トレーニング) = 0.8733333333333333
: 800. 正答率(テスト) = 0.6893
Generation: 810. 正答率(トレーニング) = 0.87
: 810. 正答率(テスト) = 0.6878
Generation: 820. 正答率(トレーニング) = 0.8666666666666667
: 820. 正答率(テスト) = 0.6929
Generation: 830. 正答率(トレーニング) = 0.8666666666666667
: 830. 正答率(テスト) = 0.688

Generation: 840. 正答率(トレーニング) = 0.8766666666666667
: 840. 正答率(テスト) = 0.6855
Generation: 850. 正答率(トレーニング) = 0.8733333333333333
: 850. 正答率(テスト) = 0.6884
Generation: 860. 正答率(トレーニング) = 0.8766666666666667
: 860. 正答率(テスト) = 0.6958
Generation: 870. 正答率(トレーニング) = 0.8766666666666667
: 870. 正答率(テスト) = 0.6951
Generation: 880. 正答率(トレーニング) = 0.8733333333333333
: 880. 正答率(テスト) = 0.6927
Generation: 890. 正答率(トレーニング) = 0.8733333333333333
: 890. 正答率(テスト) = 0.693
Generation: 900. 正答率(トレーニング) = 0.8666666666666667
: 900. 正答率(テスト) = 0.6926
Generation: 910. 正答率(トレーニング) = 0.8733333333333333
: 910. 正答率(テスト) = 0.6826
Generation: 920. 正答率(トレーニング) = 0.8766666666666667
: 920. 正答率(テスト) = 0.6817
Generation: 930. 正答率(トレーニング) = 0.8733333333333333
: 930. 正答率(テスト) = 0.6918
Generation: 940. 正答率(トレーニング) = 0.87
: 940. 正答率(テスト) = 0.695
Generation: 950. 正答率(トレーニング) = 0.8733333333333333
: 950. 正答率(テスト) = 0.6975
Generation: 960. 正答率(トレーニング) = 0.8833333333333333
: 960. 正答率(テスト) = 0.689
Generation: 970. 正答率(トレーニング) = 0.8833333333333333
: 970. 正答率(テスト) = 0.6838
Generation: 980. 正答率(トレーニング) = 0.8733333333333333
: 980. 正答率(テスト) = 0.6938
Generation: 990. 正答率(トレーニング) = 0.8733333333333333
: 990. 正答率(テスト) = 0.6967
Generation: 1000. 正答率(トレーニング) = 0.8766666666666667
: 1000. 正答率(テスト) = 0.6882

**L1**

```
In [47]: (x_train, d_train), (x_test, d_test) = load_mnist(normalize=True)

print("データ読み込み完了")

# 過学習を再現するために、学習データを削減
x_train = x_train[:300]
d_train = d_train[:300]

network = MultiLayerNet(input_size=784, hidden_size_list=[100, 100, 100, 100, 100, 100], output_size=10)

iters_num = 1000
train_size = x_train.shape[0]
batch_size = 100
learning_rate=0.1

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

plot_interval=10
hidden_layer_num = network.hidden_layer_num

# 正則化強度設定 =====
weight_decay_lambda = 0.005
# =====

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]

    grad = network.gradient(x_batch, d_batch)
    weight_decay = 0

    for idx in range(1, hidden_layer_num+1):
        grad['W' + str(idx)] = network.layers['Affine' + str(idx)].dW + weight_decay_lambda * np.sign(network.params['W' + str(idx)])
    ])
    grad['b' + str(idx)] = network.layers['Affine' + str(idx)].db
    network.params['W' + str(idx)] -= learning_rate * grad['W' + str(idx)]
    network.params['b' + str(idx)] -= learning_rate * grad['b' + str(idx)]
```

```
weight_decay += weight_decay_lambda * np.sum(np.abs(network.params['W' + str(idx)]))

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

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

    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

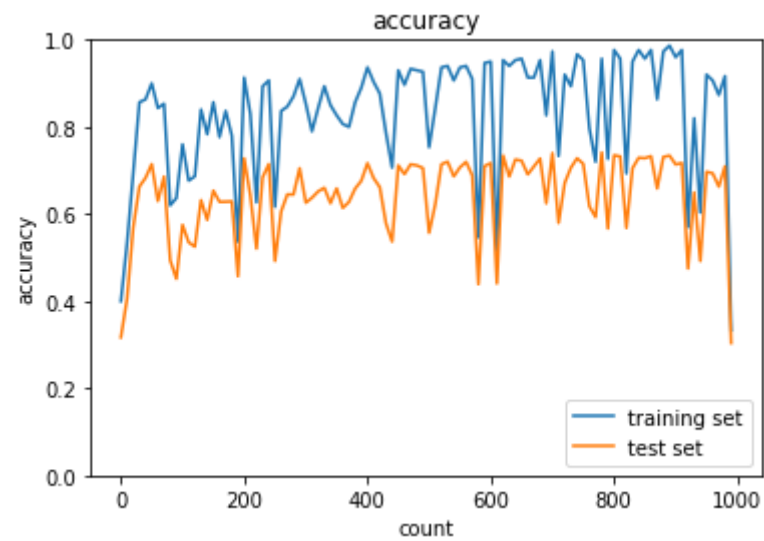
Generation: 10. 正答率(トレーニング) = 0.4
: 10. 正答率(テスト) = 0.3169
Generation: 20. 正答率(トレーニング) = 0.5333333333333333
: 20. 正答率(テスト) = 0.4055
Generation: 30. 正答率(トレーニング) = 0.6933333333333334
: 30. 正答率(テスト) = 0.5687
Generation: 40. 正答率(トレーニング) = 0.8566666666666667
: 40. 正答率(テスト) = 0.663
Generation: 50. 正答率(トレーニング) = 0.8633333333333333
: 50. 正答率(テスト) = 0.6831
Generation: 60. 正答率(トレーニング) = 0.9
: 60. 正答率(テスト) = 0.7146
Generation: 70. 正答率(トレーニング) = 0.8433333333333334
: 70. 正答率(テスト) = 0.63
Generation: 80. 正答率(トレーニング) = 0.8533333333333334
: 80. 正答率(テスト) = 0.686
Generation: 90. 正答率(トレーニング) = 0.62
: 90. 正答率(テスト) = 0.4938
Generation: 100. 正答率(トレーニング) = 0.6366666666666667
: 100. 正答率(テスト) = 0.4514
Generation: 110. 正答率(トレーニング) = 0.76
: 110. 正答率(テスト) = 0.5751
Generation: 120. 正答率(トレーニング) = 0.6766666666666666
: 120. 正答率(テスト) = 0.5351
Generation: 130. 正答率(トレーニング) = 0.6866666666666666
: 130. 正答率(テスト) = 0.5251
Generation: 140. 正答率(トレーニング) = 0.84
: 140. 正答率(テスト) = 0.632
Generation: 150. 正答率(トレーニング) = 0.7833333333333333
: 150. 正答率(テスト) = 0.5867
Generation: 160. 正答率(トレーニング) = 0.8566666666666667
: 160. 正答率(テスト) = 0.6537
Generation: 170. 正答率(トレーニング) = 0.7766666666666666
: 170. 正答率(テスト) = 0.6281
Generation: 180. 正答率(トレーニング) = 0.8366666666666667
: 180. 正答率(テスト) = 0.6284
Generation: 190. 正答率(トレーニング) = 0.78
: 190. 正答率(テスト) = 0.6291
Generation: 200. 正答率(トレーニング) = 0.5366666666666666
: 200. 正答率(テスト) = 0.4573

Generation: 210. 正答率(トレーニング) = 0.9133333333333333
: 210. 正答率(テスト) = 0.7279
Generation: 220. 正答率(トレーニング) = 0.83
: 220. 正答率(テスト) = 0.6429
Generation: 230. 正答率(トレーニング) = 0.6266666666666667
: 230. 正答率(テスト) = 0.5206
Generation: 240. 正答率(トレーニング) = 0.8933333333333333
: 240. 正答率(テスト) = 0.6845
Generation: 250. 正答率(トレーニング) = 0.9066666666666666
: 250. 正答率(テスト) = 0.7148
Generation: 260. 正答率(トレーニング) = 0.6166666666666667
: 260. 正答率(テスト) = 0.4922
Generation: 270. 正答率(トレーニング) = 0.8366666666666667
: 270. 正答率(テスト) = 0.6059
Generation: 280. 正答率(トレーニング) = 0.8466666666666667
: 280. 正答率(テスト) = 0.6451
Generation: 290. 正答率(トレーニング) = 0.87
: 290. 正答率(テスト) = 0.6448
Generation: 300. 正答率(トレーニング) = 0.91
: 300. 正答率(テスト) = 0.7052
Generation: 310. 正答率(トレーニング) = 0.8533333333333334
: 310. 正答率(テスト) = 0.626
Generation: 320. 正答率(トレーニング) = 0.79
: 320. 正答率(テスト) = 0.6374
Generation: 330. 正答率(トレーニング) = 0.8433333333333334
: 330. 正答率(テスト) = 0.6519
Generation: 340. 正答率(トレーニング) = 0.8933333333333333
: 340. 正答率(テスト) = 0.6602
Generation: 350. 正答率(トレーニング) = 0.85
: 350. 正答率(テスト) = 0.6249
Generation: 360. 正答率(トレーニング) = 0.8266666666666667
: 360. 正答率(テスト) = 0.6588
Generation: 370. 正答率(トレーニング) = 0.8066666666666666
: 370. 正答率(テスト) = 0.614
Generation: 380. 正答率(トレーニング) = 0.8
: 380. 正答率(テスト) = 0.6279
Generation: 390. 正答率(トレーニング) = 0.8566666666666667
: 390. 正答率(テスト) = 0.6584
Generation: 400. 正答率(トレーニング) = 0.89
: 400. 正答率(テスト) = 0.6763
Generation: 410. 正答率(トレーニング) = 0.9366666666666666
: 410. 正答率(テスト) = 0.7168

Generation: 420. 正答率(トレーニング) = 0.9033333333333333
: 420. 正答率(テスト) = 0.6825
Generation: 430. 正答率(トレーニング) = 0.8766666666666667
: 430. 正答率(テスト) = 0.6617
Generation: 440. 正答率(トレーニング) = 0.7866666666666666
: 440. 正答率(テスト) = 0.5789
Generation: 450. 正答率(トレーニング) = 0.7066666666666667
: 450. 正答率(テスト) = 0.5368
Generation: 460. 正答率(トレーニング) = 0.93
: 460. 正答率(テスト) = 0.7117
Generation: 470. 正答率(トレーニング) = 0.8966666666666666
: 470. 正答率(テスト) = 0.6917
Generation: 480. 正答率(トレーニング) = 0.9333333333333333
: 480. 正答率(テスト) = 0.7142
Generation: 490. 正答率(トレーニング) = 0.93
: 490. 正答率(テスト) = 0.7117
Generation: 500. 正答率(トレーニング) = 0.9266666666666666
: 500. 正答率(テスト) = 0.7056
Generation: 510. 正答率(トレーニング) = 0.7533333333333333
: 510. 正答率(テスト) = 0.5569
Generation: 520. 正答率(トレーニング) = 0.8433333333333334
: 520. 正答率(テスト) = 0.6228
Generation: 530. 正答率(トレーニング) = 0.9366666666666666
: 530. 正答率(テスト) = 0.7143
Generation: 540. 正答率(トレーニング) = 0.94
: 540. 正答率(テスト) = 0.7202
Generation: 550. 正答率(トレーニング) = 0.9066666666666666
: 550. 正答率(テスト) = 0.6864
Generation: 560. 正答率(トレーニング) = 0.9366666666666666
: 560. 正答率(テスト) = 0.7071
Generation: 570. 正答率(トレーニング) = 0.94
: 570. 正答率(テスト) = 0.7198
Generation: 580. 正答率(トレーニング) = 0.91
: 580. 正答率(テスト) = 0.6887
Generation: 590. 正答率(トレーニング) = 0.5466666666666666
: 590. 正答率(テスト) = 0.439
Generation: 600. 正答率(トレーニング) = 0.9466666666666667
: 600. 正答率(テスト) = 0.7102
Generation: 610. 正答率(トレーニング) = 0.95
: 610. 正答率(テスト) = 0.7169
Generation: 620. 正答率(トレーニング) = 0.5166666666666667
: 620. 正答率(テスト) = 0.4406

Generation: 630. 正答率(トレーニング) = 0.9533333333333334
: 630. 正答率(テスト) = 0.7343
Generation: 640. 正答率(トレーニング) = 0.94
: 640. 正答率(テスト) = 0.6861
Generation: 650. 正答率(トレーニング) = 0.9533333333333334
: 650. 正答率(テスト) = 0.7251
Generation: 660. 正答率(トレーニング) = 0.9566666666666667
: 660. 正答率(テスト) = 0.7225
Generation: 670. 正答率(トレーニング) = 0.9133333333333333
: 670. 正答率(テスト) = 0.691
Generation: 680. 正答率(トレーニング) = 0.9133333333333333
: 680. 正答率(テスト) = 0.7093
Generation: 690. 正答率(トレーニング) = 0.9533333333333334
: 690. 正答率(テスト) = 0.7274
Generation: 700. 正答率(トレーニング) = 0.8266666666666667
: 700. 正答率(テスト) = 0.6247
Generation: 710. 正答率(トレーニング) = 0.9733333333333334
: 710. 正答率(テスト) = 0.7399
Generation: 720. 正答率(トレーニング) = 0.7333333333333333
: 720. 正答率(テスト) = 0.5797
Generation: 730. 正答率(トレーニング) = 0.92
: 730. 正答率(テスト) = 0.6717
Generation: 740. 正答率(トレーニング) = 0.8933333333333333
: 740. 正答率(テスト) = 0.7065
Generation: 750. 正答率(トレーニング) = 0.9666666666666667
: 750. 正答率(テスト) = 0.728
Generation: 760. 正答率(トレーニング) = 0.9533333333333334
: 760. 正答率(テスト) = 0.7154
Generation: 770. 正答率(トレーニング) = 0.7966666666666666
: 770. 正答率(テスト) = 0.6176
Generation: 780. 正答率(トレーニング) = 0.72
: 780. 正答率(テスト) = 0.5931
Generation: 790. 正答率(トレーニング) = 0.9566666666666667
: 790. 正答率(テスト) = 0.7414
Generation: 800. 正答率(トレーニング) = 0.7266666666666667
: 800. 正答率(テスト) = 0.5668
Generation: 810. 正答率(トレーニング) = 0.9766666666666667
: 810. 正答率(テスト) = 0.7349
Generation: 820. 正答率(トレーニング) = 0.9566666666666667
: 820. 正答率(テスト) = 0.7323
Generation: 830. 正答率(トレーニング) = 0.6933333333333334
: 830. 正答率(テスト) = 0.5681

Generation: 840. 正答率(トレーニング) = 0.95
: 840. 正答率(テスト) = 0.7054
Generation: 850. 正答率(トレーニング) = 0.9766666666666667
: 850. 正答率(テスト) = 0.7292
Generation: 860. 正答率(トレーニング) = 0.9566666666666667
: 860. 正答率(テスト) = 0.7283
Generation: 870. 正答率(トレーニング) = 0.9766666666666667
: 870. 正答率(テスト) = 0.7329
Generation: 880. 正答率(トレーニング) = 0.8633333333333333
: 880. 正答率(テスト) = 0.6587
Generation: 890. 正答率(トレーニング) = 0.9733333333333334
: 890. 正答率(テスト) = 0.7311
Generation: 900. 正答率(トレーニング) = 0.9866666666666667
: 900. 正答率(テスト) = 0.7342
Generation: 910. 正答率(トレーニング) = 0.96
: 910. 正答率(テスト) = 0.7139
Generation: 920. 正答率(トレーニング) = 0.9766666666666667
: 920. 正答率(テスト) = 0.7172
Generation: 930. 正答率(トレーニング) = 0.57
: 930. 正答率(テスト) = 0.4749
Generation: 940. 正答率(トレーニング) = 0.82
: 940. 正答率(テスト) = 0.6496
Generation: 950. 正答率(トレーニング) = 0.6033333333333334
: 950. 正答率(テスト) = 0.4921
Generation: 960. 正答率(トレーニング) = 0.92
: 960. 正答率(テスト) = 0.697
Generation: 970. 正答率(トレーニング) = 0.9066666666666666
: 970. 正答率(テスト) = 0.6936
Generation: 980. 正答率(トレーニング) = 0.8733333333333333
: 980. 正答率(テスト) = 0.663
Generation: 990. 正答率(トレーニング) = 0.9166666666666666
: 990. 正答率(テスト) = 0.7096
Generation: 1000. 正答率(トレーニング) = 0.3333333333333333
: 1000. 正答率(テスト) = 0.3041



[try] weight_decay_lambdaの値を変更して正則化の強さを確認しよう

```
In [4]: from common import optimizer

(x_train, d_train), (x_test, d_test) = load_mnist(normalize=True)

print("データ読み込み完了")

# 過学習を再現するために、学習データを削減
x_train = x_train[:300]
d_train = d_train[:300]

network = MultiLayerNet(input_size=784, hidden_size_list=[100, 100, 100, 100, 100, 100], output_size=10)

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
hidden_layer_num = network.hidden_layer_num

# 正則化強度設定 =====
weight_decay_lambda = 1
# =====

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]

    grad = network.gradient(x_batch, d_batch)
    weight_decay = 0

    for idx in range(1, hidden_layer_num+1):
        grad['W' + str(idx)] = network.layers['Affine' + str(idx)].dW + weight_decay_lambda * network.params['W' + str(idx)]
        grad['b' + str(idx)] = network.layers['Affine' + str(idx)].db
```

```
network.params['W' + str(idx)] -= learning_rate * grad['W' + str(idx)]
network.params['b' + str(idx)] -= learning_rate * grad['b' + str(idx)]
weight_decay += 0.5 * weight_decay_lambda * np.sqrt(np.sum(network.params['W' + str(idx)] ** 2))

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

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

    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

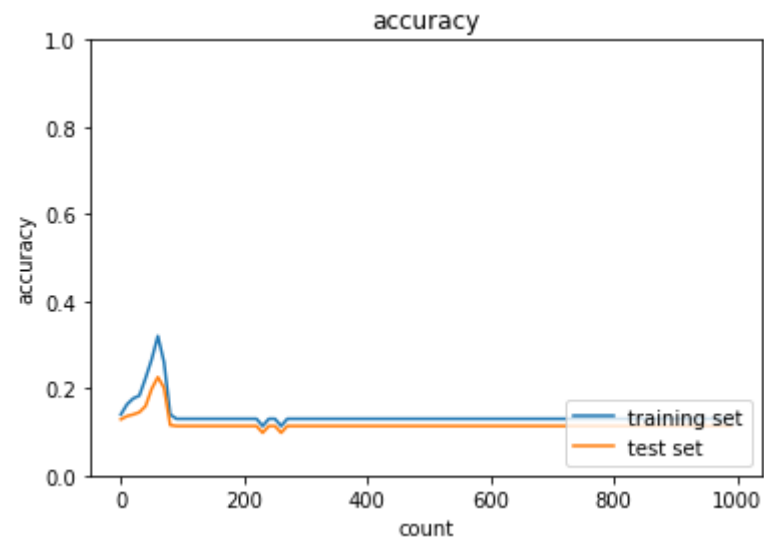
Generation: 10. 正答率(トレーニング) = 0.14
: 10. 正答率(テスト) = 0.1289
Generation: 20. 正答率(トレーニング) = 0.16333333333333333
: 20. 正答率(テスト) = 0.1362
Generation: 30. 正答率(トレーニング) = 0.17666666666666667
: 30. 正答率(テスト) = 0.1399
Generation: 40. 正答率(トレーニング) = 0.18333333333333332
: 40. 正答率(テスト) = 0.1451
Generation: 50. 正答率(トレーニング) = 0.22333333333333333
: 50. 正答率(テスト) = 0.1591
Generation: 60. 正答率(トレーニング) = 0.26666666666666666
: 60. 正答率(テスト) = 0.199
Generation: 70. 正答率(トレーニング) = 0.32
: 70. 正答率(テスト) = 0.2257
Generation: 80. 正答率(トレーニング) = 0.26
: 80. 正答率(テスト) = 0.2013
Generation: 90. 正答率(トレーニング) = 0.14
: 90. 正答率(テスト) = 0.1155
Generation: 100. 正答率(トレーニング) = 0.13
: 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) = 0.13
: 120. 正答率(テスト) = 0.1135
Generation: 130. 正答率(トレーニング) = 0.13
: 130. 正答率(テスト) = 0.1135
Generation: 140. 正答率(トレーニング) = 0.13
: 140. 正答率(テスト) = 0.1135
Generation: 150. 正答率(トレーニング) = 0.13
: 150. 正答率(テスト) = 0.1135
Generation: 160. 正答率(トレーニング) = 0.13
: 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) = 0.13
: 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) = 0.13
: 200. 正答率(テスト) = 0.1135

Generation: 210. 正答率(トレーニング) = 0.13
: 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) = 0.13
: 230. 正答率(テスト) = 0.1135
Generation: 240. 正答率(トレーニング) = 0.11333333333333333
: 240. 正答率(テスト) = 0.098
Generation: 250. 正答率(トレーニング) = 0.13
: 250. 正答率(テスト) = 0.1135
Generation: 260. 正答率(トレーニング) = 0.13
: 260. 正答率(テスト) = 0.1135
Generation: 270. 正答率(トレーニング) = 0.11333333333333333
: 270. 正答率(テスト) = 0.098
Generation: 280. 正答率(トレーニング) = 0.13
: 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) = 0.13
: 300. 正答率(テスト) = 0.1135
Generation: 310. 正答率(トレーニング) = 0.13
: 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) = 0.13
: 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) = 0.13
: 350. 正答率(テスト) = 0.1135
Generation: 360. 正答率(トレーニング) = 0.13
: 360. 正答率(テスト) = 0.1135
Generation: 370. 正答率(トレーニング) = 0.13
: 370. 正答率(テスト) = 0.1135
Generation: 380. 正答率(トレーニング) = 0.13
: 380. 正答率(テスト) = 0.1135
Generation: 390. 正答率(トレーニング) = 0.13
: 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) = 0.13
: 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) = 0.13
: 430. 正答率(テスト) = 0.1135
Generation: 440. 正答率(トレーニング) = 0.13
: 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) = 0.13
: 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) = 0.13
: 480. 正答率(テスト) = 0.1135
Generation: 490. 正答率(トレーニング) = 0.13
: 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) = 0.13
: 520. 正答率(テスト) = 0.1135
Generation: 530. 正答率(トレーニング) = 0.13
: 530. 正答率(テスト) = 0.1135
Generation: 540. 正答率(トレーニング) = 0.13
: 540. 正答率(テスト) = 0.1135
Generation: 550. 正答率(トレーニング) = 0.13
: 550. 正答率(テスト) = 0.1135
Generation: 560. 正答率(トレーニング) = 0.13
: 560. 正答率(テスト) = 0.1135
Generation: 570. 正答率(トレーニング) = 0.13
: 570. 正答率(テスト) = 0.1135
Generation: 580. 正答率(トレーニング) = 0.13
: 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) = 0.13
: 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) = 0.13
: 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) = 0.13
: 640. 正答率(テスト) = 0.1135
Generation: 650. 正答率(トレーニング) = 0.13
: 650. 正答率(テスト) = 0.1135
Generation: 660. 正答率(トレーニング) = 0.13
: 660. 正答率(テスト) = 0.1135
Generation: 670. 正答率(トレーニング) = 0.13
: 670. 正答率(テスト) = 0.1135
Generation: 680. 正答率(トレーニング) = 0.13
: 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) = 0.13
: 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) = 0.13
: 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) = 0.13
: 740. 正答率(テスト) = 0.1135
Generation: 750. 正答率(トレーニング) = 0.13
: 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) = 0.13
: 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) = 0.1135
Generation: 790. 正答率(トレーニング) = 0.13
: 790. 正答率(テスト) = 0.1135
Generation: 800. 正答率(トレーニング) = 0.13
: 800. 正答率(テスト) = 0.1135
Generation: 810. 正答率(トレーニング) = 0.13
: 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) = 0.13
: 830. 正答率(テスト) = 0.1135

Generation: 840. 正答率(トレーニング) = 0.13
: 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) = 0.13
: 860. 正答率(テスト) = 0.1135
Generation: 870. 正答率(トレーニング) = 0.13
: 870. 正答率(テスト) = 0.1135
Generation: 880. 正答率(トレーニング) = 0.13
: 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) = 0.13
: 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) = 0.13
: 940. 正答率(テスト) = 0.1135
Generation: 950. 正答率(トレーニング) = 0.13
: 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) = 0.13
: 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) = 0.13
: 990. 正答率(テスト) = 0.1135
Generation: 1000. 正答率(トレーニング) = 0.13
: 1000. 正答率(テスト) = 0.1135



Dropout

```
In [5]: class Dropout:
        def __init__(self, dropout_ratio=0.5):
            self.dropout_ratio = dropout_ratio
            self.mask = None

        def forward(self, x, train_flg=True):
            if train_flg:
                self.mask = np.random.rand(*x.shape) > self.dropout_ratio
                return x * self.mask
            else:
                return x * (1.0 - self.dropout_ratio)

        def backward(self, dout):
            return dout * self.mask
```

```
In [69]: from common import optimizer
(x_train, d_train), (x_test, d_test) = load_mnist(normalize=True)

print("データ読み込み完了")

# 過学習を再現するために、学習データを削減
x_train = x_train[:300]
d_train = d_train[:300]

# ドロップアウト設定 =====
use_dropout = True
dropout_ratio = 0.15
# =====

network = MultiLayerNet(input_size=784, hidden_size_list=[100, 100, 100, 100, 100, 100], output_size=10,
                        weight_decay_lambda=weight_decay_lambda, use_dropout = use_dropout, dropout_ratio = dropout_ratio)
optimizer = optimizer.SGD(learning_rate=0.01)
# optimizer = optimizer.Momentum(learning_rate=0.01, momentum=0.9)
# optimizer = optimizer.AdaGrad(learning_rate=0.01)
# optimizer = optimizer.Adam()

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

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]

    grad = network.gradient(x_batch, d_batch)
    optimizer.update(network.params, grad)

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

```
train_loss_list.append(loss)

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

    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

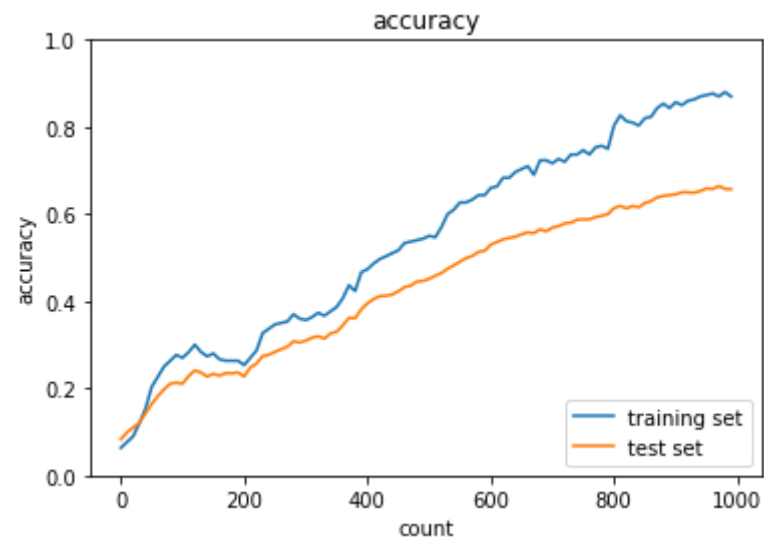
Generation: 10. 正答率(トレーニング) = 0.06333333333333334
: 10. 正答率(テスト) = 0.0833
Generation: 20. 正答率(トレーニング) = 0.07666666666666666
: 20. 正答率(テスト) = 0.0988
Generation: 30. 正答率(トレーニング) = 0.09
: 30. 正答率(テスト) = 0.1102
Generation: 40. 正答率(トレーニング) = 0.12
: 40. 正答率(テスト) = 0.1206
Generation: 50. 正答率(トレーニング) = 0.15333333333333332
: 50. 正答率(テスト) = 0.1426
Generation: 60. 正答率(トレーニング) = 0.20333333333333334
: 60. 正答率(テスト) = 0.1642
Generation: 70. 正答率(トレーニング) = 0.22666666666666666
: 70. 正答率(テスト) = 0.1821
Generation: 80. 正答率(トレーニング) = 0.25
: 80. 正答率(テスト) = 0.1974
Generation: 90. 正答率(トレーニング) = 0.2633333333333333
: 90. 正答率(テスト) = 0.2105
Generation: 100. 正答率(トレーニング) = 0.27666666666666667
: 100. 正答率(テスト) = 0.2131
Generation: 110. 正答率(トレーニング) = 0.27
: 110. 正答率(テスト) = 0.2106
Generation: 120. 正答率(トレーニング) = 0.2833333333333333
: 120. 正答率(テスト) = 0.2283
Generation: 130. 正答率(トレーニング) = 0.3
: 130. 正答率(テスト) = 0.2411
Generation: 140. 正答率(トレーニング) = 0.2833333333333333
: 140. 正答率(テスト) = 0.2364
Generation: 150. 正答率(トレーニング) = 0.2733333333333333
: 150. 正答率(テスト) = 0.227
Generation: 160. 正答率(トレーニング) = 0.28
: 160. 正答率(テスト) = 0.2333
Generation: 170. 正答率(トレーニング) = 0.26666666666666666
: 170. 正答率(テスト) = 0.2291
Generation: 180. 正答率(トレーニング) = 0.2633333333333333
: 180. 正答率(テスト) = 0.2349
Generation: 190. 正答率(トレーニング) = 0.2633333333333333
: 190. 正答率(テスト) = 0.2344
Generation: 200. 正答率(トレーニング) = 0.2633333333333333
: 200. 正答率(テスト) = 0.2365

Generation: 210. 正答率(トレーニング) = 0.25333333333333335
: 210. 正答率(テスト) = 0.2273
Generation: 220. 正答率(トレーニング) = 0.27
: 220. 正答率(テスト) = 0.2477
Generation: 230. 正答率(トレーニング) = 0.28666666666666667
: 230. 正答率(テスト) = 0.2567
Generation: 240. 正答率(トレーニング) = 0.32666666666666666
: 240. 正答率(テスト) = 0.2739
Generation: 250. 正答率(トレーニング) = 0.33666666666666667
: 250. 正答率(テスト) = 0.277
Generation: 260. 正答率(トレーニング) = 0.34666666666666667
: 260. 正答率(テスト) = 0.2836
Generation: 270. 正答率(トレーニング) = 0.35
: 270. 正答率(テスト) = 0.2897
Generation: 280. 正答率(トレーニング) = 0.35333333333333333
: 280. 正答率(テスト) = 0.2955
Generation: 290. 正答率(トレーニング) = 0.37
: 290. 正答率(テスト) = 0.3078
Generation: 300. 正答率(トレーニング) = 0.36
: 300. 正答率(テスト) = 0.3053
Generation: 310. 正答率(トレーニング) = 0.35666666666666667
: 310. 正答率(テスト) = 0.3093
Generation: 320. 正答率(トレーニング) = 0.36333333333333334
: 320. 正答率(テスト) = 0.3163
Generation: 330. 正答率(トレーニング) = 0.37333333333333335
: 330. 正答率(テスト) = 0.3195
Generation: 340. 正答率(トレーニング) = 0.36666666666666664
: 340. 正答率(テスト) = 0.3141
Generation: 350. 正答率(トレーニング) = 0.37666666666666665
: 350. 正答率(テスト) = 0.3258
Generation: 360. 正答率(トレーニング) = 0.38666666666666666
: 360. 正答率(テスト) = 0.3299
Generation: 370. 正答率(トレーニング) = 0.40666666666666667
: 370. 正答率(テスト) = 0.3442
Generation: 380. 正答率(トレーニング) = 0.43666666666666665
: 380. 正答率(テスト) = 0.3618
Generation: 390. 正答率(トレーニング) = 0.42333333333333334
: 390. 正答率(テスト) = 0.3609
Generation: 400. 正答率(トレーニング) = 0.46666666666666667
: 400. 正答率(テスト) = 0.3815
Generation: 410. 正答率(トレーニング) = 0.47333333333333333
: 410. 正答率(テスト) = 0.3952

Generation: 420. 正答率(トレーニング) = 0.4866666666666667
: 420. 正答率(テスト) = 0.4049
Generation: 430. 正答率(トレーニング) = 0.4966666666666665
: 430. 正答率(テスト) = 0.4118
Generation: 440. 正答率(トレーニング) = 0.5033333333333333
: 440. 正答率(テスト) = 0.4123
Generation: 450. 正答率(トレーニング) = 0.51
: 450. 正答率(テスト) = 0.4153
Generation: 460. 正答率(トレーニング) = 0.5166666666666667
: 460. 正答率(テスト) = 0.4226
Generation: 470. 正答率(トレーニング) = 0.5333333333333333
: 470. 正答率(テスト) = 0.4324
Generation: 480. 正答率(トレーニング) = 0.5366666666666666
: 480. 正答率(テスト) = 0.4356
Generation: 490. 正答率(トレーニング) = 0.54
: 490. 正答率(テスト) = 0.4448
Generation: 500. 正答率(トレーニング) = 0.5433333333333333
: 500. 正答率(テスト) = 0.4465
Generation: 510. 正答率(トレーニング) = 0.55
: 510. 正答率(テスト) = 0.4512
Generation: 520. 正答率(トレーニング) = 0.5466666666666666
: 520. 正答率(テスト) = 0.4585
Generation: 530. 正答率(トレーニング) = 0.57
: 530. 正答率(テスト) = 0.4646
Generation: 540. 正答率(トレーニング) = 0.6
: 540. 正答率(テスト) = 0.4746
Generation: 550. 正答率(トレーニング) = 0.61
: 550. 正答率(テスト) = 0.4822
Generation: 560. 正答率(トレーニング) = 0.6266666666666667
: 560. 正答率(テスト) = 0.4908
Generation: 570. 正答率(トレーニング) = 0.6266666666666667
: 570. 正答率(テスト) = 0.4988
Generation: 580. 正答率(トレーニング) = 0.6333333333333333
: 580. 正答率(テスト) = 0.5041
Generation: 590. 正答率(トレーニング) = 0.6433333333333333
: 590. 正答率(テスト) = 0.5133
Generation: 600. 正答率(トレーニング) = 0.6433333333333333
: 600. 正答率(テスト) = 0.5157
Generation: 610. 正答率(トレーニング) = 0.66
: 610. 正答率(テスト) = 0.5298
Generation: 620. 正答率(トレーニング) = 0.6633333333333333
: 620. 正答率(テスト) = 0.536

Generation: 630. 正答率(トレーニング) = 0.6833333333333333
: 630. 正答率(テスト) = 0.5418
Generation: 640. 正答率(トレーニング) = 0.6833333333333333
: 640. 正答率(テスト) = 0.5453
Generation: 650. 正答率(トレーニング) = 0.6966666666666667
: 650. 正答率(テスト) = 0.5481
Generation: 660. 正答率(トレーニング) = 0.7033333333333334
: 660. 正答率(テスト) = 0.5535
Generation: 670. 正答率(トレーニング) = 0.71
: 670. 正答率(テスト) = 0.5582
Generation: 680. 正答率(トレーニング) = 0.69
: 680. 正答率(テスト) = 0.5562
Generation: 690. 正答率(トレーニング) = 0.7233333333333334
: 690. 正答率(テスト) = 0.5648
Generation: 700. 正答率(トレーニング) = 0.7233333333333334
: 700. 正答率(テスト) = 0.5605
Generation: 710. 正答率(トレーニング) = 0.7166666666666667
: 710. 正答率(テスト) = 0.5689
Generation: 720. 正答率(トレーニング) = 0.7266666666666667
: 720. 正答率(テスト) = 0.5723
Generation: 730. 正答率(トレーニング) = 0.72
: 730. 正答率(テスト) = 0.5787
Generation: 740. 正答率(トレーニング) = 0.7366666666666667
: 740. 正答率(テスト) = 0.5802
Generation: 750. 正答率(トレーニング) = 0.7366666666666667
: 750. 正答率(テスト) = 0.5869
Generation: 760. 正答率(トレーニング) = 0.7466666666666667
: 760. 正答率(テスト) = 0.5879
Generation: 770. 正答率(トレーニング) = 0.7366666666666667
: 770. 正答率(テスト) = 0.5878
Generation: 780. 正答率(トレーニング) = 0.7533333333333333
: 780. 正答率(テスト) = 0.5931
Generation: 790. 正答率(トレーニング) = 0.7566666666666667
: 790. 正答率(テスト) = 0.5961
Generation: 800. 正答率(トレーニング) = 0.75
: 800. 正答率(テスト) = 0.6
Generation: 810. 正答率(トレーニング) = 0.8033333333333333
: 810. 正答率(テスト) = 0.6136
Generation: 820. 正答率(トレーニング) = 0.8266666666666667
: 820. 正答率(テスト) = 0.6186
Generation: 830. 正答率(トレーニング) = 0.8133333333333334
: 830. 正答率(テスト) = 0.6132

Generation: 840. 正答率(トレーニング) = 0.81
: 840. 正答率(テスト) = 0.6183
Generation: 850. 正答率(トレーニング) = 0.8033333333333333
: 850. 正答率(テスト) = 0.6156
Generation: 860. 正答率(トレーニング) = 0.82
: 860. 正答率(テスト) = 0.6254
Generation: 870. 正答率(トレーニング) = 0.8233333333333334
: 870. 正答率(テスト) = 0.63
Generation: 880. 正答率(トレーニング) = 0.8433333333333334
: 880. 正答率(テスト) = 0.6384
Generation: 890. 正答率(トレーニング) = 0.8533333333333334
: 890. 正答率(テスト) = 0.6417
Generation: 900. 正答率(トレーニング) = 0.8433333333333334
: 900. 正答率(テスト) = 0.6436
Generation: 910. 正答率(トレーニング) = 0.8566666666666667
: 910. 正答率(テスト) = 0.6452
Generation: 920. 正答率(トレーニング) = 0.85
: 920. 正答率(テスト) = 0.6501
Generation: 930. 正答率(トレーニング) = 0.86
: 930. 正答率(テスト) = 0.6499
Generation: 940. 正答率(トレーニング) = 0.8633333333333333
: 940. 正答率(テスト) = 0.6489
Generation: 950. 正答率(トレーニング) = 0.87
: 950. 正答率(テスト) = 0.653
Generation: 960. 正答率(トレーニング) = 0.8733333333333333
: 960. 正答率(テスト) = 0.6588
Generation: 970. 正答率(トレーニング) = 0.8766666666666667
: 970. 正答率(テスト) = 0.6576
Generation: 980. 正答率(トレーニング) = 0.87
: 980. 正答率(テスト) = 0.6641
Generation: 990. 正答率(トレーニング) = 0.88
: 990. 正答率(テスト) = 0.658
Generation: 1000. 正答率(トレーニング) = 0.87
: 1000. 正答率(テスト) = 0.6571



[try] dropout_ratioの値を変更してみよう

SGD dropout_ratio=0.5

```
In [8]: from common import optimizer
        (x_train, d_train), (x_test, d_test) = load_mnist(normalize=True)

        print("データ読み込み完了")

        # 過学習を再現するために、学習データを削減
        x_train = x_train[:300]
        d_train = d_train[:300]

        # ドロップアウト設定 =====
        use_dropout = True
        dropout_ratio = 0.5
        # =====

        network = MultiLayerNet(input_size=784, hidden_size_list=[100, 100, 100, 100, 100, 100], output_size=10,
                                weight_decay_lambda=weight_decay_lambda, use_dropout = use_dropout, dropout_ratio = dropout_ratio)
        optimizer = optimizer.SGD(learning_rate=0.01)
        # optimizer = optimizer.Momentum(learning_rate=0.01, momentum=0.9)
        # optimizer = optimizer.AdaGrad(learning_rate=0.01)
        # optimizer = optimizer.Adam()

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

        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]

            grad = network.gradient(x_batch, d_batch)
            optimizer.update(network.params, grad)

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

```
train_loss_list.append(loss)

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

    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

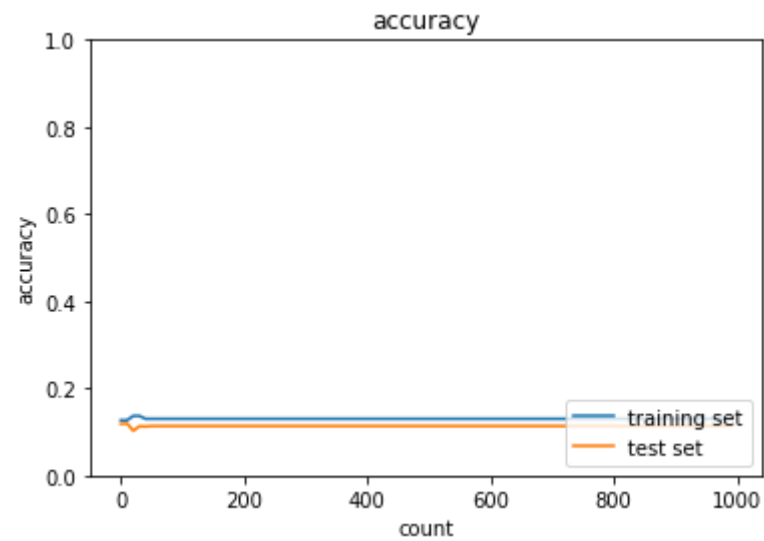
Generation: 10. 正答率(トレーニング) = 0.12666666666666668
: 10. 正答率(テスト) = 0.1185
Generation: 20. 正答率(トレーニング) = 0.12666666666666668
: 20. 正答率(テスト) = 0.1193
Generation: 30. 正答率(トレーニング) = 0.13666666666666666
: 30. 正答率(テスト) = 0.1027
Generation: 40. 正答率(トレーニング) = 0.13666666666666666
: 40. 正答率(テスト) = 0.1128
Generation: 50. 正答率(トレーニング) = 0.13
: 50. 正答率(テスト) = 0.1123
Generation: 60. 正答率(トレーニング) = 0.13
: 60. 正答率(テスト) = 0.1135
Generation: 70. 正答率(トレーニング) = 0.13
: 70. 正答率(テスト) = 0.1135
Generation: 80. 正答率(トレーニング) = 0.13
: 80. 正答率(テスト) = 0.1135
Generation: 90. 正答率(トレーニング) = 0.13
: 90. 正答率(テスト) = 0.1135
Generation: 100. 正答率(トレーニング) = 0.13
: 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) = 0.13
: 120. 正答率(テスト) = 0.1135
Generation: 130. 正答率(トレーニング) = 0.13
: 130. 正答率(テスト) = 0.1135
Generation: 140. 正答率(トレーニング) = 0.13
: 140. 正答率(テスト) = 0.1135
Generation: 150. 正答率(トレーニング) = 0.13
: 150. 正答率(テスト) = 0.1135
Generation: 160. 正答率(トレーニング) = 0.13
: 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) = 0.13
: 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) = 0.13
: 200. 正答率(テスト) = 0.1135

Generation: 210. 正答率(トレーニング) = 0.13
: 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) = 0.13
: 230. 正答率(テスト) = 0.1135
Generation: 240. 正答率(トレーニング) = 0.13
: 240. 正答率(テスト) = 0.1135
Generation: 250. 正答率(トレーニング) = 0.13
: 250. 正答率(テスト) = 0.1135
Generation: 260. 正答率(トレーニング) = 0.13
: 260. 正答率(テスト) = 0.1135
Generation: 270. 正答率(トレーニング) = 0.13
: 270. 正答率(テスト) = 0.1135
Generation: 280. 正答率(トレーニング) = 0.13
: 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) = 0.13
: 300. 正答率(テスト) = 0.1135
Generation: 310. 正答率(トレーニング) = 0.13
: 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) = 0.13
: 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) = 0.13
: 350. 正答率(テスト) = 0.1135
Generation: 360. 正答率(トレーニング) = 0.13
: 360. 正答率(テスト) = 0.1135
Generation: 370. 正答率(トレーニング) = 0.13
: 370. 正答率(テスト) = 0.1135
Generation: 380. 正答率(トレーニング) = 0.13
: 380. 正答率(テスト) = 0.1135
Generation: 390. 正答率(トレーニング) = 0.13
: 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) = 0.13
: 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) = 0.13
: 430. 正答率(テスト) = 0.1135
Generation: 440. 正答率(トレーニング) = 0.13
: 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) = 0.13
: 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) = 0.13
: 480. 正答率(テスト) = 0.1135
Generation: 490. 正答率(トレーニング) = 0.13
: 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) = 0.13
: 520. 正答率(テスト) = 0.1135
Generation: 530. 正答率(トレーニング) = 0.13
: 530. 正答率(テスト) = 0.1135
Generation: 540. 正答率(トレーニング) = 0.13
: 540. 正答率(テスト) = 0.1135
Generation: 550. 正答率(トレーニング) = 0.13
: 550. 正答率(テスト) = 0.1135
Generation: 560. 正答率(トレーニング) = 0.13
: 560. 正答率(テスト) = 0.1135
Generation: 570. 正答率(トレーニング) = 0.13
: 570. 正答率(テスト) = 0.1135
Generation: 580. 正答率(トレーニング) = 0.13
: 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) = 0.13
: 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) = 0.13
: 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) = 0.13
: 640. 正答率(テスト) = 0.1135
Generation: 650. 正答率(トレーニング) = 0.13
: 650. 正答率(テスト) = 0.1135
Generation: 660. 正答率(トレーニング) = 0.13
: 660. 正答率(テスト) = 0.1135
Generation: 670. 正答率(トレーニング) = 0.13
: 670. 正答率(テスト) = 0.1135
Generation: 680. 正答率(トレーニング) = 0.13
: 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) = 0.13
: 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) = 0.13
: 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) = 0.13
: 740. 正答率(テスト) = 0.1135
Generation: 750. 正答率(トレーニング) = 0.13
: 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) = 0.13
: 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) = 0.1135
Generation: 790. 正答率(トレーニング) = 0.13
: 790. 正答率(テスト) = 0.1135
Generation: 800. 正答率(トレーニング) = 0.13
: 800. 正答率(テスト) = 0.1135
Generation: 810. 正答率(トレーニング) = 0.13
: 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) = 0.13
: 830. 正答率(テスト) = 0.1135

Generation: 840. 正答率(トレーニング) = 0.13
: 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) = 0.13
: 860. 正答率(テスト) = 0.1135
Generation: 870. 正答率(トレーニング) = 0.13
: 870. 正答率(テスト) = 0.1135
Generation: 880. 正答率(トレーニング) = 0.13
: 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) = 0.13
: 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) = 0.13
: 940. 正答率(テスト) = 0.1135
Generation: 950. 正答率(トレーニング) = 0.13
: 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) = 0.13
: 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) = 0.13
: 990. 正答率(テスト) = 0.1135
Generation: 1000. 正答率(トレーニング) = 0.13
: 1000. 正答率(テスト) = 0.1135



[try] optimizerとdropout_ratioの値を変更してみよう

Momentum dropout_ratio=0.5

```
In [9]: from common import optimizer
        (x_train, d_train), (x_test, d_test) = load_mnist(normalize=True)

        print("データ読み込み完了")

        # 過学習を再現するために、学習データを削減
        x_train = x_train[:300]
        d_train = d_train[:300]

        # ドロップアウト設定 =====
        use_dropout = True
        dropout_ratio = 0.5
        # =====

        network = MultiLayerNet(input_size=784, hidden_size_list=[100, 100, 100, 100, 100, 100], output_size=10,
                                weight_decay_lambda=weight_decay_lambda, use_dropout = use_dropout, dropout_ratio = dropout_ratio)
        #optimizer = optimizer.SGD(learning_rate=0.01)
        optimizer = optimizer.Momentum(learning_rate=0.01, momentum=0.9)
        # optimizer = optimizer.AdaGrad(learning_rate=0.01)
        # optimizer = optimizer.Adam()

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

        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]

            grad = network.gradient(x_batch, d_batch)
            optimizer.update(network.params, grad)

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

```
train_loss_list.append(loss)

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

    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

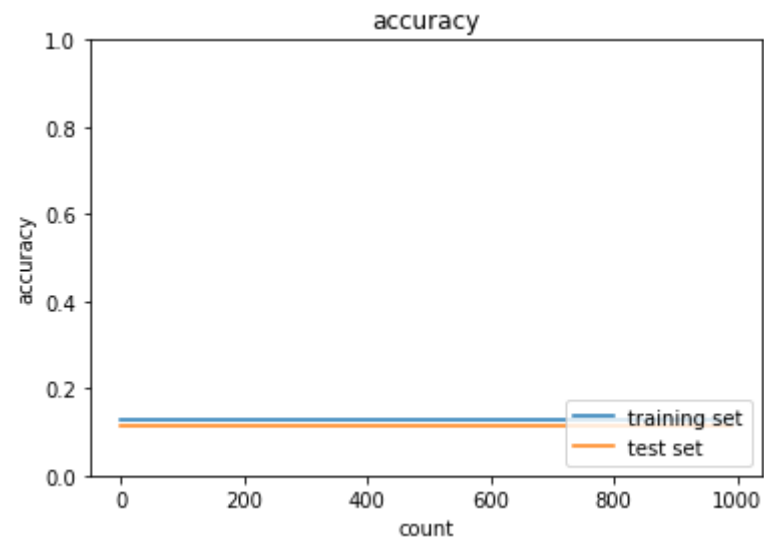
Generation: 10. 正答率(トレーニング) = 0.13
: 10. 正答率(テスト) = 0.1135
Generation: 20. 正答率(トレーニング) = 0.13
: 20. 正答率(テスト) = 0.1135
Generation: 30. 正答率(トレーニング) = 0.13
: 30. 正答率(テスト) = 0.1135
Generation: 40. 正答率(トレーニング) = 0.13
: 40. 正答率(テスト) = 0.1135
Generation: 50. 正答率(トレーニング) = 0.13
: 50. 正答率(テスト) = 0.1135
Generation: 60. 正答率(トレーニング) = 0.13
: 60. 正答率(テスト) = 0.1135
Generation: 70. 正答率(トレーニング) = 0.13
: 70. 正答率(テスト) = 0.1135
Generation: 80. 正答率(トレーニング) = 0.13
: 80. 正答率(テスト) = 0.1135
Generation: 90. 正答率(トレーニング) = 0.13
: 90. 正答率(テスト) = 0.1135
Generation: 100. 正答率(トレーニング) = 0.13
: 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) = 0.13
: 120. 正答率(テスト) = 0.1135
Generation: 130. 正答率(トレーニング) = 0.13
: 130. 正答率(テスト) = 0.1135
Generation: 140. 正答率(トレーニング) = 0.13
: 140. 正答率(テスト) = 0.1135
Generation: 150. 正答率(トレーニング) = 0.13
: 150. 正答率(テスト) = 0.1135
Generation: 160. 正答率(トレーニング) = 0.13
: 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) = 0.13
: 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) = 0.13
: 200. 正答率(テスト) = 0.1135

Generation: 210. 正答率(トレーニング) = 0.13
: 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) = 0.13
: 230. 正答率(テスト) = 0.1135
Generation: 240. 正答率(トレーニング) = 0.13
: 240. 正答率(テスト) = 0.1135
Generation: 250. 正答率(トレーニング) = 0.13
: 250. 正答率(テスト) = 0.1135
Generation: 260. 正答率(トレーニング) = 0.13
: 260. 正答率(テスト) = 0.1135
Generation: 270. 正答率(トレーニング) = 0.13
: 270. 正答率(テスト) = 0.1135
Generation: 280. 正答率(トレーニング) = 0.13
: 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) = 0.13
: 300. 正答率(テスト) = 0.1135
Generation: 310. 正答率(トレーニング) = 0.13
: 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) = 0.13
: 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) = 0.13
: 350. 正答率(テスト) = 0.1135
Generation: 360. 正答率(トレーニング) = 0.13
: 360. 正答率(テスト) = 0.1135
Generation: 370. 正答率(トレーニング) = 0.13
: 370. 正答率(テスト) = 0.1135
Generation: 380. 正答率(トレーニング) = 0.13
: 380. 正答率(テスト) = 0.1135
Generation: 390. 正答率(トレーニング) = 0.13
: 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) = 0.13
: 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) = 0.13
: 430. 正答率(テスト) = 0.1135
Generation: 440. 正答率(トレーニング) = 0.13
: 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) = 0.13
: 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) = 0.13
: 480. 正答率(テスト) = 0.1135
Generation: 490. 正答率(トレーニング) = 0.13
: 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) = 0.13
: 520. 正答率(テスト) = 0.1135
Generation: 530. 正答率(トレーニング) = 0.13
: 530. 正答率(テスト) = 0.1135
Generation: 540. 正答率(トレーニング) = 0.13
: 540. 正答率(テスト) = 0.1135
Generation: 550. 正答率(トレーニング) = 0.13
: 550. 正答率(テスト) = 0.1135
Generation: 560. 正答率(トレーニング) = 0.13
: 560. 正答率(テスト) = 0.1135
Generation: 570. 正答率(トレーニング) = 0.13
: 570. 正答率(テスト) = 0.1135
Generation: 580. 正答率(トレーニング) = 0.13
: 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) = 0.13
: 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) = 0.13
: 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) = 0.13
: 640. 正答率(テスト) = 0.1135
Generation: 650. 正答率(トレーニング) = 0.13
: 650. 正答率(テスト) = 0.1135
Generation: 660. 正答率(トレーニング) = 0.13
: 660. 正答率(テスト) = 0.1135
Generation: 670. 正答率(トレーニング) = 0.13
: 670. 正答率(テスト) = 0.1135
Generation: 680. 正答率(トレーニング) = 0.13
: 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) = 0.13
: 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) = 0.13
: 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) = 0.13
: 740. 正答率(テスト) = 0.1135
Generation: 750. 正答率(トレーニング) = 0.13
: 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) = 0.13
: 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) = 0.1135
Generation: 790. 正答率(トレーニング) = 0.13
: 790. 正答率(テスト) = 0.1135
Generation: 800. 正答率(トレーニング) = 0.13
: 800. 正答率(テスト) = 0.1135
Generation: 810. 正答率(トレーニング) = 0.13
: 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) = 0.13
: 830. 正答率(テスト) = 0.1135

Generation: 840. 正答率(トレーニング) = 0.13
: 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) = 0.13
: 860. 正答率(テスト) = 0.1135
Generation: 870. 正答率(トレーニング) = 0.13
: 870. 正答率(テスト) = 0.1135
Generation: 880. 正答率(トレーニング) = 0.13
: 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) = 0.13
: 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) = 0.13
: 940. 正答率(テスト) = 0.1135
Generation: 950. 正答率(トレーニング) = 0.13
: 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) = 0.13
: 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) = 0.13
: 990. 正答率(テスト) = 0.1135
Generation: 1000. 正答率(トレーニング) = 0.13
: 1000. 正答率(テスト) = 0.1135



[try] optimizerとdropout_ratioの値を変更してみよう

AdaGrad dropout_ratio=0.5

```
In [12]: from common import optimizer
(x_train, d_train), (x_test, d_test) = load_mnist(normalize=True)

print("データ読み込み完了")

# 過学習を再現するために、学習データを削減
x_train = x_train[:300]
d_train = d_train[:300]

# ドロップアウト設定 =====
use_dropout = True
dropout_ratio = 0.5
# =====

network = MultiLayerNet(input_size=784, hidden_size_list=[100, 100, 100, 100, 100, 100], output_size=10,
                        weight_decay_lambda=weight_decay_lambda, use_dropout = use_dropout, dropout_ratio = dropout_ratio)
#optimizer = optimizer.SGD(learning_rate=0.01)
#optimizer = optimizer.Momentum(learning_rate=0.01, momentum=0.9)
optimizer = optimizer.AdaGrad(learning_rate=0.01)
# optimizer = optimizer.Adam()

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

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]

    grad = network.gradient(x_batch, d_batch)
    optimizer.update(network.params, grad)

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

```
train_loss_list.append(loss)

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

    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

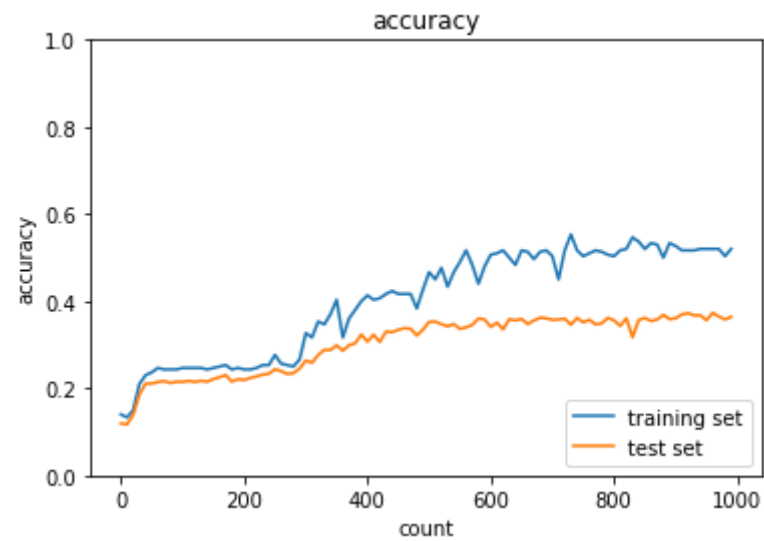
Generation: 10. 正答率(トレーニング) = 0.14
: 10. 正答率(テスト) = 0.1193
Generation: 20. 正答率(トレーニング) = 0.13333333333333333
: 20. 正答率(テスト) = 0.1173
Generation: 30. 正答率(トレーニング) = 0.15
: 30. 正答率(テスト) = 0.1394
Generation: 40. 正答率(トレーニング) = 0.21
: 40. 正答率(テスト) = 0.1856
Generation: 50. 正答率(トレーニング) = 0.23
: 50. 正答率(テスト) = 0.2107
Generation: 60. 正答率(トレーニング) = 0.23666666666666666
: 60. 正答率(テスト) = 0.211
Generation: 70. 正答率(トレーニング) = 0.24666666666666667
: 70. 正答率(テスト) = 0.2143
Generation: 80. 正答率(トレーニング) = 0.24333333333333335
: 80. 正答率(テスト) = 0.2167
Generation: 90. 正答率(トレーニング) = 0.24333333333333335
: 90. 正答率(テスト) = 0.2123
Generation: 100. 正答率(トレーニング) = 0.24333333333333335
: 100. 正答率(テスト) = 0.2154
Generation: 110. 正答率(トレーニング) = 0.24666666666666667
: 110. 正答率(テスト) = 0.2147
Generation: 120. 正答率(トレーニング) = 0.24666666666666667
: 120. 正答率(テスト) = 0.217
Generation: 130. 正答率(トレーニング) = 0.24666666666666667
: 130. 正答率(テスト) = 0.2146
Generation: 140. 正答率(トレーニング) = 0.24666666666666667
: 140. 正答率(テスト) = 0.2174
Generation: 150. 正答率(トレーニング) = 0.24333333333333335
: 150. 正答率(テスト) = 0.2152
Generation: 160. 正答率(トレーニング) = 0.24666666666666667
: 160. 正答率(テスト) = 0.2212
Generation: 170. 正答率(トレーニング) = 0.25
: 170. 正答率(テスト) = 0.2255
Generation: 180. 正答率(トレーニング) = 0.25333333333333335
: 180. 正答率(テスト) = 0.2303
Generation: 190. 正答率(トレーニング) = 0.24333333333333335
: 190. 正答率(テスト) = 0.2156
Generation: 200. 正答率(トレーニング) = 0.24666666666666667
: 200. 正答率(テスト) = 0.2208

Generation: 210. 正答率(トレーニング) = 0.2433333333333335
: 210. 正答率(テスト) = 0.2193
Generation: 220. 正答率(トレーニング) = 0.2433333333333335
: 220. 正答率(テスト) = 0.2237
Generation: 230. 正答率(トレーニング) = 0.2466666666666667
: 230. 正答率(テスト) = 0.2275
Generation: 240. 正答率(トレーニング) = 0.2533333333333335
: 240. 正答率(テスト) = 0.2315
Generation: 250. 正答率(トレーニング) = 0.2533333333333335
: 250. 正答率(テスト) = 0.2334
Generation: 260. 正答率(トレーニング) = 0.2766666666666667
: 260. 正答率(テスト) = 0.244
Generation: 270. 正答率(トレーニング) = 0.2566666666666665
: 270. 正答率(テスト) = 0.2394
Generation: 280. 正答率(トレーニング) = 0.2533333333333335
: 280. 正答率(テスト) = 0.2331
Generation: 290. 正答率(トレーニング) = 0.25
: 290. 正答率(テスト) = 0.2349
Generation: 300. 正答率(トレーニング) = 0.2666666666666666
: 300. 正答率(テスト) = 0.2454
Generation: 310. 正答率(トレーニング) = 0.3266666666666666
: 310. 正答率(テスト) = 0.2638
Generation: 320. 正答率(トレーニング) = 0.3166666666666665
: 320. 正答率(テスト) = 0.2586
Generation: 330. 正答率(トレーニング) = 0.3533333333333333
: 330. 正答率(テスト) = 0.2764
Generation: 340. 正答率(トレーニング) = 0.3466666666666667
: 340. 正答率(テスト) = 0.2883
Generation: 350. 正答率(トレーニング) = 0.37
: 350. 正答率(テスト) = 0.288
Generation: 360. 正答率(トレーニング) = 0.4033333333333333
: 360. 正答率(テスト) = 0.2986
Generation: 370. 正答率(トレーニング) = 0.3166666666666665
: 370. 正答率(テスト) = 0.2861
Generation: 380. 正答率(トレーニング) = 0.36
: 380. 正答率(テスト) = 0.2989
Generation: 390. 正答率(トレーニング) = 0.38
: 390. 正答率(テスト) = 0.3033
Generation: 400. 正答率(トレーニング) = 0.4
: 400. 正答率(テスト) = 0.3235
Generation: 410. 正答率(トレーニング) = 0.4133333333333333
: 410. 正答率(テスト) = 0.3072

Generation: 420. 正答率(トレーニング) = 0.4033333333333333
: 420. 正答率(テスト) = 0.323
Generation: 430. 正答率(トレーニング) = 0.4066666666666667
: 430. 正答率(テスト) = 0.3064
Generation: 440. 正答率(トレーニング) = 0.4166666666666667
: 440. 正答率(テスト) = 0.3303
Generation: 450. 正答率(トレーニング) = 0.42333333333333334
: 450. 正答率(テスト) = 0.3287
Generation: 460. 正答率(トレーニング) = 0.4166666666666667
: 460. 正答率(テスト) = 0.3343
Generation: 470. 正答率(トレーニング) = 0.4166666666666667
: 470. 正答率(テスト) = 0.3381
Generation: 480. 正答率(トレーニング) = 0.4166666666666667
: 480. 正答率(テスト) = 0.3368
Generation: 490. 正答率(トレーニング) = 0.38333333333333336
: 490. 正答率(テスト) = 0.3212
Generation: 500. 正答率(トレーニング) = 0.4266666666666667
: 500. 正答率(テスト) = 0.3347
Generation: 510. 正答率(トレーニング) = 0.4666666666666667
: 510. 正答率(テスト) = 0.3523
Generation: 520. 正答率(トレーニング) = 0.45
: 520. 正答率(テスト) = 0.353
Generation: 530. 正答率(トレーニング) = 0.4766666666666667
: 530. 正答率(テスト) = 0.3473
Generation: 540. 正答率(トレーニング) = 0.43333333333333335
: 540. 正答率(テスト) = 0.3424
Generation: 550. 正答率(トレーニング) = 0.4666666666666667
: 550. 正答率(テスト) = 0.3474
Generation: 560. 正答率(トレーニング) = 0.49
: 560. 正答率(テスト) = 0.3364
Generation: 570. 正答率(トレーニング) = 0.5166666666666667
: 570. 正答率(テスト) = 0.34
Generation: 580. 正答率(トレーニング) = 0.48333333333333334
: 580. 正答率(テスト) = 0.3456
Generation: 590. 正答率(トレーニング) = 0.44
: 590. 正答率(テスト) = 0.3603
Generation: 600. 正答率(トレーニング) = 0.48
: 600. 正答率(テスト) = 0.3582
Generation: 610. 正答率(トレーニング) = 0.5066666666666667
: 610. 正答率(テスト) = 0.3409
Generation: 620. 正答率(トレーニング) = 0.51
: 620. 正答率(テスト) = 0.3507

Generation: 630. 正答率(トレーニング) = 0.5166666666666667
: 630. 正答率(テスト) = 0.3353
Generation: 640. 正答率(トレーニング) = 0.5
: 640. 正答率(テスト) = 0.3586
Generation: 650. 正答率(トレーニング) = 0.48333333333333334
: 650. 正答率(テスト) = 0.3558
Generation: 660. 正答率(トレーニング) = 0.5166666666666667
: 660. 正答率(テスト) = 0.3595
Generation: 670. 正答率(トレーニング) = 0.5133333333333333
: 670. 正答率(テスト) = 0.3476
Generation: 680. 正答率(トレーニング) = 0.49666666666666665
: 680. 正答率(テスト) = 0.3559
Generation: 690. 正答率(トレーニング) = 0.5133333333333333
: 690. 正答率(テスト) = 0.362
Generation: 700. 正答率(トレーニング) = 0.5166666666666667
: 700. 正答率(テスト) = 0.3608
Generation: 710. 正答率(トレーニング) = 0.5033333333333333
: 710. 正答率(テスト) = 0.3573
Generation: 720. 正答率(トレーニング) = 0.45
: 720. 正答率(テスト) = 0.3579
Generation: 730. 正答率(トレーニング) = 0.5166666666666667
: 730. 正答率(テスト) = 0.3599
Generation: 740. 正答率(トレーニング) = 0.5533333333333333
: 740. 正答率(テスト) = 0.3461
Generation: 750. 正答率(トレーニング) = 0.5166666666666667
: 750. 正答率(テスト) = 0.3616
Generation: 760. 正答率(トレーニング) = 0.5033333333333333
: 760. 正答率(テスト) = 0.3515
Generation: 770. 正答率(トレーニング) = 0.51
: 770. 正答率(テスト) = 0.3573
Generation: 780. 正答率(トレーニング) = 0.5166666666666667
: 780. 正答率(テスト) = 0.3467
Generation: 790. 正答率(トレーニング) = 0.5133333333333333
: 790. 正答率(テスト) = 0.3493
Generation: 800. 正答率(トレーニング) = 0.5066666666666667
: 800. 正答率(テスト) = 0.3618
Generation: 810. 正答率(トレーニング) = 0.5033333333333333
: 810. 正答率(テスト) = 0.3559
Generation: 820. 正答率(トレーニング) = 0.5166666666666667
: 820. 正答率(テスト) = 0.3434
Generation: 830. 正答率(トレーニング) = 0.52
: 830. 正答率(テスト) = 0.3609

Generation: 840. 正答率(トレーニング) = 0.5466666666666666
: 840. 正答率(テスト) = 0.3171
Generation: 850. 正答率(トレーニング) = 0.5366666666666666
: 850. 正答率(テスト) = 0.3569
Generation: 860. 正答率(トレーニング) = 0.52
: 860. 正答率(テスト) = 0.3617
Generation: 870. 正答率(トレーニング) = 0.5333333333333333
: 870. 正答率(テスト) = 0.3546
Generation: 880. 正答率(トレーニング) = 0.53
: 880. 正答率(テスト) = 0.358
Generation: 890. 正答率(トレーニング) = 0.5
: 890. 正答率(テスト) = 0.3686
Generation: 900. 正答率(トレーニング) = 0.5333333333333333
: 900. 正答率(テスト) = 0.3589
Generation: 910. 正答率(トレーニング) = 0.5266666666666666
: 910. 正答率(テスト) = 0.3609
Generation: 920. 正答率(トレーニング) = 0.5166666666666667
: 920. 正答率(テスト) = 0.3696
Generation: 930. 正答率(トレーニング) = 0.5166666666666667
: 930. 正答率(テスト) = 0.3725
Generation: 940. 正答率(トレーニング) = 0.5166666666666667
: 940. 正答率(テスト) = 0.3674
Generation: 950. 正答率(トレーニング) = 0.52
: 950. 正答率(テスト) = 0.3675
Generation: 960. 正答率(トレーニング) = 0.52
: 960. 正答率(テスト) = 0.3565
Generation: 970. 正答率(トレーニング) = 0.52
: 970. 正答率(テスト) = 0.373
Generation: 980. 正答率(トレーニング) = 0.52
: 980. 正答率(テスト) = 0.3649
Generation: 990. 正答率(トレーニング) = 0.5033333333333333
: 990. 正答率(テスト) = 0.3582
Generation: 1000. 正答率(トレーニング) = 0.52
: 1000. 正答率(テスト) = 0.364



[try] optimizerとdropout_ratioの値を変更してみよう

Adam dropout_ratio=0.5

```
In [11]: from common import optimizer
(x_train, d_train), (x_test, d_test) = load_mnist(normalize=True)

print("データ読み込み完了")

# 過学習を再現するために、学習データを削減
x_train = x_train[:300]
d_train = d_train[:300]

# ドロップアウト設定 =====
use_dropout = True
dropout_ratio = 0.5
# =====

network = MultiLayerNet(input_size=784, hidden_size_list=[100, 100, 100, 100, 100, 100], output_size=10,
                        weight_decay_lambda=weight_decay_lambda, use_dropout = use_dropout, dropout_ratio = dropout_ratio)
#optimizer = optimizer.SGD(learning_rate=0.01)
#optimizer = optimizer.Momentum(learning_rate=0.01, momentum=0.9)
#optimizer = optimizer.AdaGrad(learning_rate=0.01)
optimizer = optimizer.Adam()

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

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]

    grad = network.gradient(x_batch, d_batch)
    optimizer.update(network.params, grad)

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

```
train_loss_list.append(loss)

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

    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()
```

The history saving thread hit an unexpected error (OperationalError('database or disk is full',)).History will not be written to the database.

データ読み込み完了

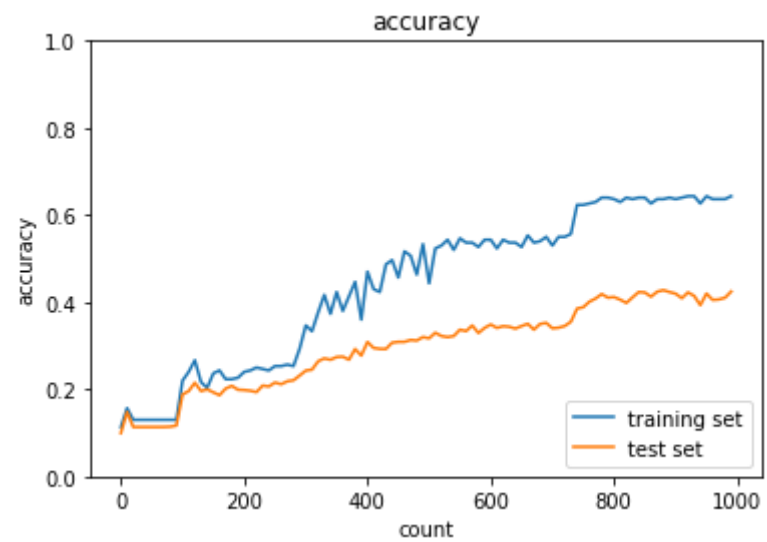
Generation: 10. 正答率(トレーニング) = 0.11333333333333333
: 10. 正答率(テスト) = 0.0997
Generation: 20. 正答率(トレーニング) = 0.15666666666666668
: 20. 正答率(テスト) = 0.148
Generation: 30. 正答率(トレーニング) = 0.13
: 30. 正答率(テスト) = 0.1136
Generation: 40. 正答率(トレーニング) = 0.13
: 40. 正答率(テスト) = 0.1135
Generation: 50. 正答率(トレーニング) = 0.13
: 50. 正答率(テスト) = 0.1135
Generation: 60. 正答率(トレーニング) = 0.13
: 60. 正答率(テスト) = 0.1135
Generation: 70. 正答率(トレーニング) = 0.13
: 70. 正答率(テスト) = 0.1135
Generation: 80. 正答率(トレーニング) = 0.13
: 80. 正答率(テスト) = 0.1136
Generation: 90. 正答率(トレーニング) = 0.13
: 90. 正答率(テスト) = 0.1142
Generation: 100. 正答率(トレーニング) = 0.13
: 100. 正答率(テスト) = 0.1168
Generation: 110. 正答率(トレーニング) = 0.22
: 110. 正答率(テスト) = 0.1882
Generation: 120. 正答率(トレーニング) = 0.24
: 120. 正答率(テスト) = 0.196
Generation: 130. 正答率(トレーニング) = 0.26666666666666666
: 130. 正答率(テスト) = 0.215
Generation: 140. 正答率(トレーニング) = 0.21666666666666667
: 140. 正答率(テスト) = 0.196
Generation: 150. 正答率(トレーニング) = 0.20333333333333334
: 150. 正答率(テスト) = 0.1999
Generation: 160. 正答率(トレーニング) = 0.23666666666666666
: 160. 正答率(テスト) = 0.193
Generation: 170. 正答率(トレーニング) = 0.24333333333333335
: 170. 正答率(テスト) = 0.1865
Generation: 180. 正答率(トレーニング) = 0.22333333333333333
: 180. 正答率(テスト) = 0.2017
Generation: 190. 正答率(トレーニング) = 0.22333333333333333
: 190. 正答率(テスト) = 0.2079

Generation: 200. 正答率(トレーニング) = 0.22666666666666666
: 200. 正答率(テスト) = 0.1991
Generation: 210. 正答率(トレーニング) = 0.24
: 210. 正答率(テスト) = 0.1983
Generation: 220. 正答率(トレーニング) = 0.24333333333333335
: 220. 正答率(テスト) = 0.1968
Generation: 230. 正答率(トレーニング) = 0.25
: 230. 正答率(テスト) = 0.1936
Generation: 240. 正答率(トレーニング) = 0.24666666666666667
: 240. 正答率(テスト) = 0.2086
Generation: 250. 正答率(トレーニング) = 0.24333333333333335
: 250. 正答率(テスト) = 0.2065
Generation: 260. 正答率(トレーニング) = 0.25333333333333335
: 260. 正答率(テスト) = 0.2155
Generation: 270. 正答率(トレーニング) = 0.25333333333333335
: 270. 正答率(テスト) = 0.2117
Generation: 280. 正答率(トレーニング) = 0.25666666666666665
: 280. 正答率(テスト) = 0.2184
Generation: 290. 正答率(トレーニング) = 0.25333333333333335
: 290. 正答率(テスト) = 0.2207
Generation: 300. 正答率(トレーニング) = 0.29333333333333333
: 300. 正答率(テスト) = 0.2316
Generation: 310. 正答率(トレーニング) = 0.34666666666666667
: 310. 正答率(テスト) = 0.243
Generation: 320. 正答率(トレーニング) = 0.3333333333333333
: 320. 正答率(テスト) = 0.2451
Generation: 330. 正答率(トレーニング) = 0.37666666666666665
: 330. 正答率(テスト) = 0.2643
Generation: 340. 正答率(トレーニング) = 0.41666666666666667
: 340. 正答率(テスト) = 0.271
Generation: 350. 正答率(トレーニング) = 0.37333333333333335
: 350. 正答率(テスト) = 0.2679
Generation: 360. 正答率(トレーニング) = 0.42333333333333334
: 360. 正答率(テスト) = 0.2741
Generation: 370. 正答率(トレーニング) = 0.38
: 370. 正答率(テスト) = 0.2749
Generation: 380. 正答率(トレーニング) = 0.41333333333333333
: 380. 正答率(テスト) = 0.268
Generation: 390. 正答率(トレーニング) = 0.44666666666666666
: 390. 正答率(テスト) = 0.2925
Generation: 400. 正答率(トレーニング) = 0.36
: 400. 正答率(テスト) = 0.2774

Generation: 410. 正答率(トレーニング) = 0.47
: 410. 正答率(テスト) = 0.3087
Generation: 420. 正答率(トレーニング) = 0.43
: 420. 正答率(テスト) = 0.2949
Generation: 430. 正答率(トレーニング) = 0.42333333333333334
: 430. 正答率(テスト) = 0.2929
Generation: 440. 正答率(トレーニング) = 0.48666666666666667
: 440. 正答率(テスト) = 0.2926
Generation: 450. 正答率(トレーニング) = 0.49666666666666665
: 450. 正答率(テスト) = 0.307
Generation: 460. 正答率(トレーニング) = 0.45666666666666667
: 460. 正答率(テスト) = 0.3089
Generation: 470. 正答率(トレーニング) = 0.51666666666666667
: 470. 正答率(テスト) = 0.3089
Generation: 480. 正答率(トレーニング) = 0.50666666666666667
: 480. 正答率(テスト) = 0.3132
Generation: 490. 正答率(トレーニング) = 0.46333333333333333
: 490. 正答率(テスト) = 0.3119
Generation: 500. 正答率(トレーニング) = 0.53333333333333333
: 500. 正答率(テスト) = 0.3199
Generation: 510. 正答率(トレーニング) = 0.44333333333333336
: 510. 正答率(テスト) = 0.317
Generation: 520. 正答率(トレーニング) = 0.52333333333333333
: 520. 正答率(テスト) = 0.3298
Generation: 530. 正答率(トレーニング) = 0.53
: 530. 正答率(テスト) = 0.3223
Generation: 540. 正答率(トレーニング) = 0.54333333333333333
: 540. 正答率(テスト) = 0.3199
Generation: 550. 正答率(トレーニング) = 0.52
: 550. 正答率(テスト) = 0.3223
Generation: 560. 正答率(トレーニング) = 0.54666666666666666
: 560. 正答率(テスト) = 0.3369
Generation: 570. 正答率(トレーニング) = 0.53666666666666666
: 570. 正答率(テスト) = 0.3336
Generation: 580. 正答率(トレーニング) = 0.53666666666666666
: 580. 正答率(テスト) = 0.3465
Generation: 590. 正答率(トレーニング) = 0.52666666666666666
: 590. 正答率(テスト) = 0.3293
Generation: 600. 正答率(トレーニング) = 0.54333333333333333
: 600. 正答率(テスト) = 0.341
Generation: 610. 正答率(トレーニング) = 0.54333333333333333
: 610. 正答率(テスト) = 0.3488

Generation: 620. 正答率(トレーニング) = 0.5233333333333333
: 620. 正答率(テスト) = 0.341
Generation: 630. 正答率(トレーニング) = 0.5433333333333333
: 630. 正答率(テスト) = 0.3449
Generation: 640. 正答率(トレーニング) = 0.5366666666666666
: 640. 正答率(テスト) = 0.3436
Generation: 650. 正答率(トレーニング) = 0.5366666666666666
: 650. 正答率(テスト) = 0.3394
Generation: 660. 正答率(トレーニング) = 0.5266666666666666
: 660. 正答率(テスト) = 0.3452
Generation: 670. 正答率(トレーニング) = 0.5533333333333333
: 670. 正答率(テスト) = 0.3504
Generation: 680. 正答率(トレーニング) = 0.5366666666666666
: 680. 正答率(テスト) = 0.337
Generation: 690. 正答率(トレーニング) = 0.54
: 690. 正答率(テスト) = 0.35
Generation: 700. 正答率(トレーニング) = 0.55
: 700. 正答率(テスト) = 0.3525
Generation: 710. 正答率(トレーニング) = 0.53
: 710. 正答率(テスト) = 0.3401
Generation: 720. 正答率(トレーニング) = 0.55
: 720. 正答率(テスト) = 0.3414
Generation: 730. 正答率(トレーニング) = 0.55
: 730. 正答率(テスト) = 0.3453
Generation: 740. 正答率(トレーニング) = 0.5566666666666666
: 740. 正答率(テスト) = 0.3551
Generation: 750. 正答率(トレーニング) = 0.6233333333333333
: 750. 正答率(テスト) = 0.3853
Generation: 760. 正答率(トレーニング) = 0.6233333333333333
: 760. 正答率(テスト) = 0.3884
Generation: 770. 正答率(トレーニング) = 0.6266666666666667
: 770. 正答率(テスト) = 0.4011
Generation: 780. 正答率(トレーニング) = 0.63
: 780. 正答率(テスト) = 0.4084
Generation: 790. 正答率(トレーニング) = 0.64
: 790. 正答率(テスト) = 0.4186
Generation: 800. 正答率(トレーニング) = 0.64
: 800. 正答率(テスト) = 0.4105
Generation: 810. 正答率(トレーニング) = 0.6366666666666667
: 810. 正答率(テスト) = 0.4117
Generation: 820. 正答率(トレーニング) = 0.63
: 820. 正答率(テスト) = 0.4058

Generation: 830. 正答率(トレーニング) = 0.64
: 830. 正答率(テスト) = 0.3983
Generation: 840. 正答率(トレーニング) = 0.6366666666666667
: 840. 正答率(テスト) = 0.411
Generation: 850. 正答率(トレーニング) = 0.64
: 850. 正答率(テスト) = 0.4233
Generation: 860. 正答率(トレーニング) = 0.64
: 860. 正答率(テスト) = 0.4223
Generation: 870. 正答率(トレーニング) = 0.6266666666666667
: 870. 正答率(テスト) = 0.4126
Generation: 880. 正答率(トレーニング) = 0.6366666666666667
: 880. 正答率(テスト) = 0.4241
Generation: 890. 正答率(トレーニング) = 0.6366666666666667
: 890. 正答率(テスト) = 0.4276
Generation: 900. 正答率(トレーニング) = 0.64
: 900. 正答率(テスト) = 0.4233
Generation: 910. 正答率(トレーニング) = 0.6366666666666667
: 910. 正答率(テスト) = 0.4195
Generation: 920. 正答率(トレーニング) = 0.64
: 920. 正答率(テスト) = 0.4091
Generation: 930. 正答率(トレーニング) = 0.6433333333333333
: 930. 正答率(テスト) = 0.4225
Generation: 940. 正答率(トレーニング) = 0.6433333333333333
: 940. 正答率(テスト) = 0.4144
Generation: 950. 正答率(トレーニング) = 0.6266666666666667
: 950. 正答率(テスト) = 0.3932
Generation: 960. 正答率(トレーニング) = 0.6433333333333333
: 960. 正答率(テスト) = 0.4199
Generation: 970. 正答率(トレーニング) = 0.6366666666666667
: 970. 正答率(テスト) = 0.4052
Generation: 980. 正答率(トレーニング) = 0.6366666666666667
: 980. 正答率(テスト) = 0.4063
Generation: 990. 正答率(トレーニング) = 0.6366666666666667
: 990. 正答率(テスト) = 0.4102
Generation: 1000. 正答率(トレーニング) = 0.6433333333333333
: 1000. 正答率(テスト) = 0.4245



Dropout + L1

```
In [6]: from common import optimizer
        (x_train, d_train), (x_test, d_test) = load_mnist(normalize=True)

        print("データ読み込み完了")

        # 過学習を再現するために、学習データを削減
        x_train = x_train[:300]
        d_train = d_train[:300]

        # ドロップアウト設定 =====
        use_dropout = True
        dropout_ratio = 0.15
        # =====

        network = MultiLayerNet(input_size=784, hidden_size_list=[100, 100, 100, 100, 100, 100], output_size=10,
                                use_dropout = use_dropout, dropout_ratio = dropout_ratio)

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

        train_loss_list = []
        accuracies_train = []
        accuracies_test = []
        hidden_layer_num = network.hidden_layer_num

        plot_interval=10

        # 正則化強度設定 =====
        weight_decay_lambda=0.004
        # =====

        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]

            grad = network.gradient(x_batch, d_batch)
            weight_decay = 0
```

```

for idx in range(1, hidden_layer_num+1):
    grad['W' + str(idx)] = network.layers['Affine' + str(idx)].dW + weight_decay_lambda * np.sign(network.params['W' + str(idx)
    ])
    grad['b' + str(idx)] = network.layers['Affine' + str(idx)].db
    network.params['W' + str(idx)] -= learning_rate * grad['W' + str(idx)]
    network.params['b' + str(idx)] -= learning_rate * grad['b' + str(idx)]
    weight_decay += weight_decay_lambda * np.sum(np.abs(network.params['W' + str(idx)]))

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

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

    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) = 0.1181
Generation: 20. 正答率(トレーニング) = 0.11333333333333333
: 20. 正答率(テスト) = 0.1255
Generation: 30. 正答率(トレーニング) = 0.13
: 30. 正答率(テスト) = 0.1266
Generation: 40. 正答率(トレーニング) = 0.14
: 40. 正答率(テスト) = 0.1295
Generation: 50. 正答率(トレーニング) = 0.15
: 50. 正答率(テスト) = 0.1318
Generation: 60. 正答率(トレーニング) = 0.14333333333333334
: 60. 正答率(テスト) = 0.1344
Generation: 70. 正答率(トレーニング) = 0.14333333333333334
: 70. 正答率(テスト) = 0.1339
Generation: 80. 正答率(トレーニング) = 0.14666666666666667
: 80. 正答率(テスト) = 0.1349
Generation: 90. 正答率(トレーニング) = 0.14666666666666667
: 90. 正答率(テスト) = 0.1361
Generation: 100. 正答率(トレーニング) = 0.15
: 100. 正答率(テスト) = 0.1352
Generation: 110. 正答率(トレーニング) = 0.17333333333333334
: 110. 正答率(テスト) = 0.1385
Generation: 120. 正答率(トレーニング) = 0.18
: 120. 正答率(テスト) = 0.1411
Generation: 130. 正答率(トレーニング) = 0.17333333333333334
: 130. 正答率(テスト) = 0.1378
Generation: 140. 正答率(トレーニング) = 0.19333333333333333
: 140. 正答率(テスト) = 0.1406
Generation: 150. 正答率(トレーニング) = 0.18333333333333332
: 150. 正答率(テスト) = 0.1433
Generation: 160. 正答率(トレーニング) = 0.19333333333333333
: 160. 正答率(テスト) = 0.1477
Generation: 170. 正答率(トレーニング) = 0.21666666666666667
: 170. 正答率(テスト) = 0.1538
Generation: 180. 正答率(トレーニング) = 0.21666666666666667
: 180. 正答率(テスト) = 0.1605
Generation: 190. 正答率(トレーニング) = 0.24333333333333335
: 190. 正答率(テスト) = 0.1671
Generation: 200. 正答率(トレーニング) = 0.25
: 200. 正答率(テスト) = 0.1777

Generation: 210. 正答率(トレーニング) = 0.25
: 210. 正答率(テスト) = 0.1826
Generation: 220. 正答率(トレーニング) = 0.25666666666666665
: 220. 正答率(テスト) = 0.1916
Generation: 230. 正答率(トレーニング) = 0.26
: 230. 正答率(テスト) = 0.2035
Generation: 240. 正答率(トレーニング) = 0.28
: 240. 正答率(テスト) = 0.2189
Generation: 250. 正答率(トレーニング) = 0.2933333333333333
: 250. 正答率(テスト) = 0.2312
Generation: 260. 正答率(トレーニング) = 0.2833333333333333
: 260. 正答率(テスト) = 0.2351
Generation: 270. 正答率(トレーニング) = 0.30666666666666664
: 270. 正答率(テスト) = 0.2424
Generation: 280. 正答率(トレーニング) = 0.31
: 280. 正答率(テスト) = 0.2452
Generation: 290. 正答率(トレーニング) = 0.32
: 290. 正答率(テスト) = 0.2507
Generation: 300. 正答率(トレーニング) = 0.33
: 300. 正答率(テスト) = 0.2563
Generation: 310. 正答率(トレーニング) = 0.33
: 310. 正答率(テスト) = 0.2583
Generation: 320. 正答率(トレーニング) = 0.33
: 320. 正答率(テスト) = 0.2626
Generation: 330. 正答率(トレーニング) = 0.33
: 330. 正答率(テスト) = 0.2651
Generation: 340. 正答率(トレーニング) = 0.32666666666666666
: 340. 正答率(テスト) = 0.2633
Generation: 350. 正答率(トレーニング) = 0.32
: 350. 正答率(テスト) = 0.2613
Generation: 360. 正答率(トレーニング) = 0.31333333333333335
: 360. 正答率(テスト) = 0.2623
Generation: 370. 正答率(トレーニング) = 0.31333333333333335
: 370. 正答率(テスト) = 0.2654
Generation: 380. 正答率(トレーニング) = 0.32333333333333333
: 380. 正答率(テスト) = 0.2704
Generation: 390. 正答率(トレーニング) = 0.32
: 390. 正答率(テスト) = 0.2727
Generation: 400. 正答率(トレーニング) = 0.32
: 400. 正答率(テスト) = 0.2723
Generation: 410. 正答率(トレーニング) = 0.33666666666666667
: 410. 正答率(テスト) = 0.2804

Generation: 420. 正答率(トレーニング) = 0.32
: 420. 正答率(テスト) = 0.2773
Generation: 430. 正答率(トレーニング) = 0.33
: 430. 正答率(テスト) = 0.2813
Generation: 440. 正答率(トレーニング) = 0.35
: 440. 正答率(テスト) = 0.2885
Generation: 450. 正答率(トレーニング) = 0.34
: 450. 正答率(テスト) = 0.2854
Generation: 460. 正答率(トレーニング) = 0.3466666666666667
: 460. 正答率(テスト) = 0.2919
Generation: 470. 正答率(トレーニング) = 0.3433333333333333
: 470. 正答率(テスト) = 0.2906
Generation: 480. 正答率(トレーニング) = 0.3433333333333333
: 480. 正答率(テスト) = 0.2879
Generation: 490. 正答率(トレーニング) = 0.34
: 490. 正答率(テスト) = 0.2859
Generation: 500. 正答率(トレーニング) = 0.3533333333333333
: 500. 正答率(テスト) = 0.2904
Generation: 510. 正答率(トレーニング) = 0.34
: 510. 正答率(テスト) = 0.2851
Generation: 520. 正答率(トレーニング) = 0.3533333333333333
: 520. 正答率(テスト) = 0.2888
Generation: 530. 正答率(トレーニング) = 0.3466666666666667
: 530. 正答率(テスト) = 0.2858
Generation: 540. 正答率(トレーニング) = 0.35
: 540. 正答率(テスト) = 0.2834
Generation: 550. 正答率(トレーニング) = 0.3566666666666667
: 550. 正答率(テスト) = 0.2861
Generation: 560. 正答率(トレーニング) = 0.36
: 560. 正答率(テスト) = 0.2891
Generation: 570. 正答率(トレーニング) = 0.36
: 570. 正答率(テスト) = 0.293
Generation: 580. 正答率(トレーニング) = 0.3733333333333333
: 580. 正答率(テスト) = 0.294
Generation: 590. 正答率(トレーニング) = 0.3733333333333333
: 590. 正答率(テスト) = 0.2959
Generation: 600. 正答率(トレーニング) = 0.37
: 600. 正答率(テスト) = 0.3002
Generation: 610. 正答率(トレーニング) = 0.37
: 610. 正答率(テスト) = 0.3027
Generation: 620. 正答率(トレーニング) = 0.3733333333333333
: 620. 正答率(テスト) = 0.3061

Generation: 630. 正答率(トレーニング) = 0.37666666666666665
: 630. 正答率(テスト) = 0.31
Generation: 640. 正答率(トレーニング) = 0.37666666666666665
: 640. 正答率(テスト) = 0.3112
Generation: 650. 正答率(トレーニング) = 0.37666666666666665
: 650. 正答率(テスト) = 0.3092
Generation: 660. 正答率(トレーニング) = 0.37666666666666665
: 660. 正答率(テスト) = 0.3116
Generation: 670. 正答率(トレーニング) = 0.36666666666666664
: 670. 正答率(テスト) = 0.3101
Generation: 680. 正答率(トレーニング) = 0.36333333333333334
: 680. 正答率(テスト) = 0.3074
Generation: 690. 正答率(トレーニング) = 0.37
: 690. 正答率(テスト) = 0.3068
Generation: 700. 正答率(トレーニング) = 0.37666666666666665
: 700. 正答率(テスト) = 0.3119
Generation: 710. 正答率(トレーニング) = 0.38
: 710. 正答率(テスト) = 0.3119
Generation: 720. 正答率(トレーニング) = 0.38666666666666666
: 720. 正答率(テスト) = 0.3152
Generation: 730. 正答率(トレーニング) = 0.38666666666666666
: 730. 正答率(テスト) = 0.3181
Generation: 740. 正答率(トレーニング) = 0.38666666666666666
: 740. 正答率(テスト) = 0.3153
Generation: 750. 正答率(トレーニング) = 0.38
: 750. 正答率(テスト) = 0.3152
Generation: 760. 正答率(トレーニング) = 0.38333333333333336
: 760. 正答率(テスト) = 0.3157
Generation: 770. 正答率(トレーニング) = 0.39
: 770. 正答率(テスト) = 0.3179
Generation: 780. 正答率(トレーニング) = 0.39
: 780. 正答率(テスト) = 0.3189
Generation: 790. 正答率(トレーニング) = 0.38
: 790. 正答率(テスト) = 0.3169
Generation: 800. 正答率(トレーニング) = 0.38333333333333336
: 800. 正答率(テスト) = 0.3204
Generation: 810. 正答率(トレーニング) = 0.39333333333333333
: 810. 正答率(テスト) = 0.3216
Generation: 820. 正答率(トレーニング) = 0.4
: 820. 正答率(テスト) = 0.3283
Generation: 830. 正答率(トレーニング) = 0.40666666666666667
: 830. 正答率(テスト) = 0.3311

Generation: 840. 正答率(トレーニング) = 0.4
: 840. 正答率(テスト) = 0.3301
Generation: 850. 正答率(トレーニング) = 0.4
: 850. 正答率(テスト) = 0.3312
Generation: 860. 正答率(トレーニング) = 0.4
: 860. 正答率(テスト) = 0.3372
Generation: 870. 正答率(トレーニング) = 0.4066666666666667
: 870. 正答率(テスト) = 0.3374
Generation: 880. 正答率(トレーニング) = 0.4033333333333333
: 880. 正答率(テスト) = 0.3371
Generation: 890. 正答率(トレーニング) = 0.4066666666666667
: 890. 正答率(テスト) = 0.3441
Generation: 900. 正答率(トレーニング) = 0.4066666666666667
: 900. 正答率(テスト) = 0.338
Generation: 910. 正答率(トレーニング) = 0.41
: 910. 正答率(テスト) = 0.3407
Generation: 920. 正答率(トレーニング) = 0.4133333333333333
: 920. 正答率(テスト) = 0.349
Generation: 930. 正答率(トレーニング) = 0.42333333333333334
: 930. 正答率(テスト) = 0.3479
Generation: 940. 正答率(トレーニング) = 0.4133333333333333
: 940. 正答率(テスト) = 0.3492
Generation: 950. 正答率(トレーニング) = 0.42
: 950. 正答率(テスト) = 0.3516
Generation: 960. 正答率(トレーニング) = 0.43
: 960. 正答率(テスト) = 0.3573
Generation: 970. 正答率(トレーニング) = 0.42
: 970. 正答率(テスト) = 0.3548
Generation: 980. 正答率(トレーニング) = 0.4266666666666667
: 980. 正答率(テスト) = 0.3537
Generation: 990. 正答率(トレーニング) = 0.4133333333333333
: 990. 正答率(テスト) = 0.3505
Generation: 1000. 正答率(トレーニング) = 0.4166666666666667
: 1000. 正答率(テスト) = 0.3513

