

神经网络的“学习学”

吴汶钊

2017 年 9 月 16 日

0.1 实验方法

固定总的训练周期数量 (Epoch) 为 10, 比较如下不同的训练方式: - 先用 0 ~ 4 这 5 个数字训练卷积网络 ConvNet, 再用后面的 5 ~ 9 这五个数字训练 ConvNet; (half_half) - 先用 0 ~ 4 这 5 个数字训练卷积网络 ConvNet, 然后再用 0 ~ 9 全部十个数字来训练 ConvNet; (half_all) - 直接用 0 ~ 9 这十个数字训练 ConvNet.(all) #### 细节设计 1. 0-4 训练集和 5-9 训练集大小设置为 25000, 0-9 训练集大小设置为 25000 2. half_half 和 half_all 实验中两次训练的 epoch 分别为总 epoch 的一半 3. 实验结果存入 results 中, 包括训练集 loss (平均), 校验集 loss (平均), 校验集准确率和测试集准确率

In [104]: # 导入所需要的包

```
import torch
import torch.nn as nn
from torch.autograd import Variable
import torch.optim as optim
import torch.nn.functional as F

import torchvision.datasets as dsets
import torchvision.transforms as transforms

import matplotlib.pyplot as plt
import numpy as np

%matplotlib inline
```

In [105]: # 训练和测试数据集

```
train_dataset = dsets.MNIST(root='./data',
                             train=True,
                             transform=transforms.ToTensor(),
                             download=True)

test_dataset = dsets.MNIST(root='./data',
```

```

train=False,
transform=transforms.ToTensor())

```

In [106]: # Hyper Parameters

```

image_size = 28
num_classes = 10
num_epochs = 10 # 训练的总循环周期
batch_size = 64
learning_rate = 0.001

```

In [107]: # 训练数据三种采样器下标, 每个采样器大小都为 25000

```

Indices_train1 = []
Indices_train2 = []
for i in range(len(train_dataset)):
    data,target = train_dataset[i]
    if target <= 4:
        Indices_train1.append(i)
for j in range(len(train_dataset)):
    data,target = train_dataset[j]
    if target > 4:
        Indices_train2.append(j)
Indices_train1 = np.random.permutation(Indices_train1)[:25000]
Indices_train2 = np.random.permutation(Indices_train2)[:25000]
Indices_train = np.random.permutation(range(len(train_dataset)))[:25000]

```

In [108]: # 训练集数据采样器和加载器

```

sampler_train1 = torch.utils.data.sampler.SubsetRandomSampler(Indices_train1)
sampler_train2 = torch.utils.data.sampler.SubsetRandomSampler(Indices_train2)
sampler_train = torch.utils.data.sampler.SubsetRandomSampler(Indices_train)
train_loader1 = torch.utils.data.DataLoader(dataset=train_dataset,
                                             batch_size=batch_size,
                                             shuffle=False,
                                             sampler = sampler_train1
                                             )
train_loader2 = torch.utils.data.DataLoader(dataset=train_dataset,
                                             batch_size=batch_size,
                                             shuffle=False,
                                             sampler = sampler_train2
                                             )
train_loader = torch.utils.data.DataLoader(dataset=train_dataset,
                                             batch_size=batch_size,

```

```

        shuffle=False,
        sampler = sampler_train
    )

```

In [109]: # 测试数据集的加载器，自动将数据分割成 *batch*

```

permuter = np.random.permutation(range(len(test_dataset)))
indices_val = permuter[:5000]
indices_test = permuter[5000:]
sampler_val = torch.utils.data.sampler.SubsetRandomSampler(indices_val)
sampler_test = torch.utils.data.sampler.SubsetRandomSampler(indices_test)
validation_loader = torch.utils.data.DataLoader(dataset=train_dataset,
                                                batch_size=batch_size,
                                                shuffle=False,
                                                sampler=sampler_val
                                                )

test_loader = torch.utils.data.DataLoader(dataset=test_dataset,
                                          batch_size=batch_size,
                                          shuffle=False,
                                          sampler=sampler_test
                                          )

```

In [110]: # 定义卷积神经网络：4 和 8 为人为指定的两个卷积层的厚度

```

depth = [4, 8]
class ConvNet(nn.Module):
    def __init__(self):
        super(ConvNet, self).__init__()
        self.conv1 = nn.Conv2d(1, 4, 5, padding = 2)
        self.pool = nn.MaxPool2d(2, 2)
        self.conv2 = nn.Conv2d(depth[0], depth[1], 5, padding = 2)
        self.fc1 = nn.Linear(image_size // 4 * image_size // 4 * depth[1] , 512)
        self.fc2 = nn.Linear(512, num_classes)

    def forward(self, x):
        # 神经网络完成一步前馈运算的过程，从输入到输出
        x = F.relu(self.conv1(x))
        x = self.pool(x)
        x = F.relu(self.conv2(x))
        x = self.pool(x)
        x = x.view(-1, image_size // 4 * image_size // 4 * depth[1])
        x = F.relu(self.fc1(x))
        x = F.dropout(x, training=self.training)

```

```

        x = self.fc2(x)
        x = F.log_softmax(x)
        return x

```

In [111]: # 定义准确率函数

```

def rightness(predictions, labels):
    pred = torch.max(predictions.data, 1)[1]
    rights = pred.eq(labels.data.view_as(pred)).sum()
    return rights, len(labels)

```

In [112]: results = {}

```

times = 5
# 开始实验 1
experiment = 'half_half'
for time in range(times):
    net = ConvNet()
    criterion = nn.CrossEntropyLoss()
    optimizer = optim.SGD(net.parameters(), lr=0.001, momentum=0.9)
    record = []

    # 开始训练循环
    for epoch in range(int(num_epochs / 2)):
        losses = []
        train_rights = []
        for batch_idx, (data, target) in enumerate(train_loader1):
            data, target = Variable(data), Variable(target)
            net.train()
            output = net(data)
            loss = criterion(output, target)
            optimizer.zero_grad()
            loss.backward()
            optimizer.step()
            right = rightness(output, target)
            train_rights.append(right)
            losses.append(loss.data.numpy())

        if batch_idx % 100 == 0:
            train_r = 1.0 * np.sum([i[0] for i in train_rights]) / np.sum([i[1] for
            net.eval()
            val_rights = []

```

```

val_losses = []
for (data, target) in validation_loader:
    data, target = Variable(data), Variable(target)
    output = net(data)
    right = rightness(output, target)
    val_rights.append(right)
    val_losses.append(loss.data.numpy())
val_r = 1.0 * np.sum([i[0] for i in val_rights]) / np.sum([i[1] for i in val_rights])
record.append([np.mean(losses), np.mean(val_losses), val_r])
print('{}实验 \t第{}次 \t训练周期:{} {}/{} \t训练误差: {:.6f} \t校验误差: {:.6f}'.format(
    experiment, time, epoch, batch_idx, len(train_loader1),
    np.mean(losses), np.mean(val_losses), train_r, val_r))

for epoch in range(int(num_epochs/2 + 1), num_epochs):
    train_rights = []
    losses = []
    for batch_idx, (data, target) in enumerate(train_loader2):
        data, target = Variable(data), Variable(target)
        net.train()
        output = net(data)
        loss = criterion(output, target)
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()
        right = rightness(output, target)
        train_rights.append(right)
        losses.append(loss.data.numpy())

    if batch_idx % 100 == 0:
        train_r = 1.0 * np.sum([i[0] for i in train_rights]) / np.sum([i[1] for i in train_rights])
        net.eval()
        val_rights = []
        val_losses = []
        for (data, target) in validation_loader:
            data, target = Variable(data), Variable(target)
            output = net(data)
            right = rightness(output, target)
            val_rights.append(right)
            val_losses.append(loss.data.numpy())

```

```

val_r = 1.0 * np.sum([i[0] for i in val_rights]) / np.sum([i[1] for i in val_rights])
record.append([np.mean(losses), np.mean(val_losses), val_r])
print('{}实验 \t第{}次 \t训练周期:{} {}/{} \t训练误差: {:.6f} \t校验误差: {:.6f}'.format(
    experiment, time, epoch, batch_idx, len(train_loader2),
    np.mean(losses), np.mean(val_losses), train_r, val_r))

net.eval()
test_rights = []
test_losses = []
for (data, target) in test_loader:
    data, target = Variable(data), Variable(target)
    output = net(data)
    right = rightness(output, target)
    test_rights.append(right)
    test_losses.append(loss.data.numpy())
right_ratio = 1.0 * np.sum([i[0] for i in test_rights]) / np.sum([i[1] for i in test_rights])
results[(experiment,time)] = [record, right_ratio]

```

half_half	实验	第 0 次	训练周期:0 0/391	训练误差: 2.296699	校验误差: 2.296699
half_half	实验	第 0 次	训练周期:0 100/391	训练误差: 1.920490	校验误差: 1.920490
half_half	实验	第 0 次	训练周期:0 200/391	训练误差: 1.569984	校验误差: 1.569984
half_half	实验	第 0 次	训练周期:0 300/391	训练误差: 1.205257	校验误差: 1.205257
half_half	实验	第 0 次	训练周期:1 0/391	训练误差: 0.122517	校验误差: 0.122517
half_half	实验	第 0 次	训练周期:1 100/391	训练误差: 0.224164	校验误差: 0.224164
half_half	实验	第 0 次	训练周期:1 200/391	训练误差: 0.209475	校验误差: 0.209475
half_half	实验	第 0 次	训练周期:1 300/391	训练误差: 0.195760	校验误差: 0.195760
half_half	实验	第 0 次	训练周期:2 0/391	训练误差: 0.096430	校验误差: 0.096430
half_half	实验	第 0 次	训练周期:2 100/391	训练误差: 0.146502	校验误差: 0.146502
half_half	实验	第 0 次	训练周期:2 200/391	训练误差: 0.141478	校验误差: 0.141478
half_half	实验	第 0 次	训练周期:2 300/391	训练误差: 0.138937	校验误差: 0.138937
half_half	实验	第 0 次	训练周期:3 0/391	训练误差: 0.098174	校验误差: 0.098174
half_half	实验	第 0 次	训练周期:3 100/391	训练误差: 0.122940	校验误差: 0.122940
half_half	实验	第 0 次	训练周期:3 200/391	训练误差: 0.119281	校验误差: 0.119281
half_half	实验	第 0 次	训练周期:3 300/391	训练误差: 0.116425	校验误差: 0.116425
half_half	实验	第 0 次	训练周期:4 0/391	训练误差: 0.081017	校验误差: 0.081017
half_half	实验	第 0 次	训练周期:4 100/391	训练误差: 0.110856	校验误差: 0.110856
half_half	实验	第 0 次	训练周期:4 200/391	训练误差: 0.109010	校验误差: 0.109010
half_half	实验	第 0 次	训练周期:4 300/391	训练误差: 0.104798	校验误差: 0.104798
half_half	实验	第 0 次	训练周期:6 0/391	训练误差: 11.101466	校验误差: 11.101466
half_half	实验	第 0 次	训练周期:6 100/391	训练误差: 1.303057	校验误差: 1.303057
half_half	实验	第 0 次	训练周期:6 200/391	训练误差: 0.900660	校验误差: 0.900660

half_half	实验	第 0 次	训练周期:6 300/391	训练误差: 0.728063	校验误差:
half_half	实验	第 0 次	训练周期:7 0/391	训练误差: 0.362129	校验误差:
half_half	实验	第 0 次	训练周期:7 100/391	训练误差: 0.291931	校验误差:
half_half	实验	第 0 次	训练周期:7 200/391	训练误差: 0.275005	校验误差:
half_half	实验	第 0 次	训练周期:7 300/391	训练误差: 0.260792	校验误差:
half_half	实验	第 0 次	训练周期:8 0/391	训练误差: 0.203270	校验误差:
half_half	实验	第 0 次	训练周期:8 100/391	训练误差: 0.207824	校验误差:
half_half	实验	第 0 次	训练周期:8 200/391	训练误差: 0.202436	校验误差:
half_half	实验	第 0 次	训练周期:8 300/391	训练误差: 0.197635	校验误差:
half_half	实验	第 0 次	训练周期:9 0/391	训练误差: 0.157834	校验误差:
half_half	实验	第 0 次	训练周期:9 100/391	训练误差: 0.163252	校验误差:
half_half	实验	第 0 次	训练周期:9 200/391	训练误差: 0.164896	校验误差:
half_half	实验	第 0 次	训练周期:9 300/391	训练误差: 0.162866	校验误差:
half_half	实验	第 1 次	训练周期:0 0/391	训练误差: 2.284150	校验误差:
half_half	实验	第 1 次	训练周期:0 100/391	训练误差: 1.837106	校验误差:
half_half	实验	第 1 次	训练周期:0 200/391	训练误差: 1.402409	校验误差:
half_half	实验	第 1 次	训练周期:0 300/391	训练误差: 1.056873	校验误差:
half_half	实验	第 1 次	训练周期:1 0/391	训练误差: 0.261545	校验误差:
half_half	实验	第 1 次	训练周期:1 100/391	训练误差: 0.200247	校验误差:
half_half	实验	第 1 次	训练周期:1 200/391	训练误差: 0.191316	校验误差:
half_half	实验	第 1 次	训练周期:1 300/391	训练误差: 0.183741	校验误差:
half_half	实验	第 1 次	训练周期:2 0/391	训练误差: 0.056843	校验误差:
half_half	实验	第 1 次	训练周期:2 100/391	训练误差: 0.145942	校验误差:
half_half	实验	第 1 次	训练周期:2 200/391	训练误差: 0.139322	校验误差:
half_half	实验	第 1 次	训练周期:2 300/391	训练误差: 0.137510	校验误差:
half_half	实验	第 1 次	训练周期:3 0/391	训练误差: 0.141705	校验误差:
half_half	实验	第 1 次	训练周期:3 100/391	训练误差: 0.117678	校验误差:
half_half	实验	第 1 次	训练周期:3 200/391	训练误差: 0.114822	校验误差:
half_half	实验	第 1 次	训练周期:3 300/391	训练误差: 0.114753	校验误差:
half_half	实验	第 1 次	训练周期:4 0/391	训练误差: 0.091636	校验误差:
half_half	实验	第 1 次	训练周期:4 100/391	训练误差: 0.105747	校验误差:
half_half	实验	第 1 次	训练周期:4 200/391	训练误差: 0.102434	校验误差:
half_half	实验	第 1 次	训练周期:4 300/391	训练误差: 0.101346	校验误差:
half_half	实验	第 1 次	训练周期:6 0/391	训练误差: 11.468720	校验误差:
half_half	实验	第 1 次	训练周期:6 100/391	训练误差: 1.297297	校验误差:
half_half	实验	第 1 次	训练周期:6 200/391	训练误差: 0.883669	校验误差:
half_half	实验	第 1 次	训练周期:6 300/391	训练误差: 0.708813	校验误差:
half_half	实验	第 1 次	训练周期:7 0/391	训练误差: 0.357160	校验误差:
half_half	实验	第 1 次	训练周期:7 100/391	训练误差: 0.277335	校验误差:

half_half	实验	第 1 次	训练周期:7 200/391	训练误差: 0.267917	校验误差:
half_half	实验	第 1 次	训练周期:7 300/391	训练误差: 0.252046	校验误差:
half_half	实验	第 1 次	训练周期:8 0/391	训练误差: 0.115919	校验误差:
half_half	实验	第 1 次	训练周期:8 100/391	训练误差: 0.192869	校验误差:
half_half	实验	第 1 次	训练周期:8 200/391	训练误差: 0.190249	校验误差:
half_half	实验	第 1 次	训练周期:8 300/391	训练误差: 0.185313	校验误差:
half_half	实验	第 1 次	训练周期:9 0/391	训练误差: 0.286932	校验误差:
half_half	实验	第 1 次	训练周期:9 100/391	训练误差: 0.153082	校验误差:
half_half	实验	第 1 次	训练周期:9 200/391	训练误差: 0.158061	校验误差:
half_half	实验	第 1 次	训练周期:9 300/391	训练误差: 0.152075	校验误差:
half_half	实验	第 2 次	训练周期:0 0/391	训练误差: 2.306233	校验误差:
half_half	实验	第 2 次	训练周期:0 100/391	训练误差: 2.189868	校验误差:
half_half	实验	第 2 次	训练周期:0 200/391	训练误差: 1.925731	校验误差:
half_half	实验	第 2 次	训练周期:0 300/391	训练误差: 1.708964	校验误差:
half_half	实验	第 2 次	训练周期:1 0/391	训练误差: 0.328404	校验误差:
half_half	实验	第 2 次	训练周期:1 100/391	训练误差: 0.300006	校验误差:
half_half	实验	第 2 次	训练周期:1 200/391	训练误差: 0.262228	校验误差:
half_half	实验	第 2 次	训练周期:1 300/391	训练误差: 0.237683	校验误差:
half_half	实验	第 2 次	训练周期:2 0/391	训练误差: 0.089211	校验误差:
half_half	实验	第 2 次	训练周期:2 100/391	训练误差: 0.153830	校验误差:
half_half	实验	第 2 次	训练周期:2 200/391	训练误差: 0.152584	校验误差:
half_half	实验	第 2 次	训练周期:2 300/391	训练误差: 0.150968	校验误差:
half_half	实验	第 2 次	训练周期:3 0/391	训练误差: 0.207484	校验误差:
half_half	实验	第 2 次	训练周期:3 100/391	训练误差: 0.133572	校验误差:
half_half	实验	第 2 次	训练周期:3 200/391	训练误差: 0.127282	校验误差:
half_half	实验	第 2 次	训练周期:3 300/391	训练误差: 0.123560	校验误差:
half_half	实验	第 2 次	训练周期:4 0/391	训练误差: 0.150457	校验误差:
half_half	实验	第 2 次	训练周期:4 100/391	训练误差: 0.109662	校验误差:
half_half	实验	第 2 次	训练周期:4 200/391	训练误差: 0.108422	校验误差:
half_half	实验	第 2 次	训练周期:4 300/391	训练误差: 0.109960	校验误差:
half_half	实验	第 2 次	训练周期:6 0/391	训练误差: 10.614622	校验误差:
half_half	实验	第 2 次	训练周期:6 100/391	训练误差: 1.383293	校验误差:
half_half	实验	第 2 次	训练周期:6 200/391	训练误差: 0.955034	校验误差:
half_half	实验	第 2 次	训练周期:6 300/391	训练误差: 0.771235	校验误差:
half_half	实验	第 2 次	训练周期:7 0/391	训练误差: 0.329701	校验误差:
half_half	实验	第 2 次	训练周期:7 100/391	训练误差: 0.321847	校验误差:
half_half	实验	第 2 次	训练周期:7 200/391	训练误差: 0.301063	校验误差:
half_half	实验	第 2 次	训练周期:7 300/391	训练误差: 0.284742	校验误差:
half_half	实验	第 2 次	训练周期:8 0/391	训练误差: 0.257917	校验误差:

half_half	实验	第 2 次	训练周期:8 100/391	训练误差: 0.214608	校验误差:
half_half	实验	第 2 次	训练周期:8 200/391	训练误差: 0.214817	校验误差:
half_half	实验	第 2 次	训练周期:8 300/391	训练误差: 0.212545	校验误差:
half_half	实验	第 2 次	训练周期:9 0/391	训练误差: 0.073922	校验误差:
half_half	实验	第 2 次	训练周期:9 100/391	训练误差: 0.169747	校验误差:
half_half	实验	第 2 次	训练周期:9 200/391	训练误差: 0.174857	校验误差:
half_half	实验	第 2 次	训练周期:9 300/391	训练误差: 0.173630	校验误差:
half_half	实验	第 3 次	训练周期:0 0/391	训练误差: 2.291459	校验误差:
half_half	实验	第 3 次	训练周期:0 100/391	训练误差: 2.177199	校验误差:
half_half	实验	第 3 次	训练周期:0 200/391	训练误差: 1.834247	校验误差:
half_half	实验	第 3 次	训练周期:0 300/391	训练误差: 1.438485	校验误差:
half_half	实验	第 3 次	训练周期:1 0/391	训练误差: 0.174299	校验误差:
half_half	实验	第 3 次	训练周期:1 100/391	训练误差: 0.217332	校验误差:
half_half	实验	第 3 次	训练周期:1 200/391	训练误差: 0.205788	校验误差:
half_half	实验	第 3 次	训练周期:1 300/391	训练误差: 0.194904	校验误差:
half_half	实验	第 3 次	训练周期:2 0/391	训练误差: 0.228412	校验误差:
half_half	实验	第 3 次	训练周期:2 100/391	训练误差: 0.149285	校验误差:
half_half	实验	第 3 次	训练周期:2 200/391	训练误差: 0.139160	校验误差:
half_half	实验	第 3 次	训练周期:2 300/391	训练误差: 0.134764	校验误差:
half_half	实验	第 3 次	训练周期:3 0/391	训练误差: 0.028653	校验误差:
half_half	实验	第 3 次	训练周期:3 100/391	训练误差: 0.110690	校验误差:
half_half	实验	第 3 次	训练周期:3 200/391	训练误差: 0.118352	校验误差:
half_half	实验	第 3 次	训练周期:3 300/391	训练误差: 0.114209	校验误差:
half_half	实验	第 3 次	训练周期:4 0/391	训练误差: 0.296590	校验误差:
half_half	实验	第 3 次	训练周期:4 100/391	训练误差: 0.105191	校验误差:
half_half	实验	第 3 次	训练周期:4 200/391	训练误差: 0.099249	校验误差:
half_half	实验	第 3 次	训练周期:4 300/391	训练误差: 0.099960	校验误差:
half_half	实验	第 3 次	训练周期:6 0/391	训练误差: 11.056499	校验误差:
half_half	实验	第 3 次	训练周期:6 100/391	训练误差: 1.305060	校验误差:
half_half	实验	第 3 次	训练周期:6 200/391	训练误差: 0.887293	校验误差:
half_half	实验	第 3 次	训练周期:6 300/391	训练误差: 0.710712	校验误差:
half_half	实验	第 3 次	训练周期:7 0/391	训练误差: 0.256439	校验误差:
half_half	实验	第 3 次	训练周期:7 100/391	训练误差: 0.252390	校验误差:
half_half	实验	第 3 次	训练周期:7 200/391	训练误差: 0.242723	校验误差:
half_half	实验	第 3 次	训练周期:7 300/391	训练误差: 0.234322	校验误差:
half_half	实验	第 3 次	训练周期:8 0/391	训练误差: 0.236547	校验误差:
half_half	实验	第 3 次	训练周期:8 100/391	训练误差: 0.178155	校验误差:
half_half	实验	第 3 次	训练周期:8 200/391	训练误差: 0.175748	校验误差:
half_half	实验	第 3 次	训练周期:8 300/391	训练误差: 0.175091	校验误差:

half_half	实验	第 3 次	训练周期:9 0/391	训练误差: 0.205216	校验误差:
half_half	实验	第 3 次	训练周期:9 100/391	训练误差: 0.145703	校验误差:
half_half	实验	第 3 次	训练周期:9 200/391	训练误差: 0.148038	校验误差:
half_half	实验	第 3 次	训练周期:9 300/391	训练误差: 0.143443	校验误差:
half_half	实验	第 4 次	训练周期:0 0/391	训练误差: 2.318525	校验误差:
half_half	实验	第 4 次	训练周期:0 100/391	训练误差: 2.248976	校验误差:
half_half	实验	第 4 次	训练周期:0 200/391	训练误差: 2.110878	校验误差:
half_half	实验	第 4 次	训练周期:0 300/391	训练误差: 1.867426	校验误差:
half_half	实验	第 4 次	训练周期:1 0/391	训练误差: 0.382520	校验误差:
half_half	实验	第 4 次	训练周期:1 100/391	训练误差: 0.279282	校验误差:
half_half	实验	第 4 次	训练周期:1 200/391	训练误差: 0.244668	校验误差:
half_half	实验	第 4 次	训练周期:1 300/391	训练误差: 0.225306	校验误差:
half_half	实验	第 4 次	训练周期:2 0/391	训练误差: 0.216644	校验误差:
half_half	实验	第 4 次	训练周期:2 100/391	训练误差: 0.145615	校验误差:
half_half	实验	第 4 次	训练周期:2 200/391	训练误差: 0.142480	校验误差:
half_half	实验	第 4 次	训练周期:2 300/391	训练误差: 0.142497	校验误差:
half_half	实验	第 4 次	训练周期:3 0/391	训练误差: 0.101762	校验误差:
half_half	实验	第 4 次	训练周期:3 100/391	训练误差: 0.119104	校验误差:
half_half	实验	第 4 次	训练周期:3 200/391	训练误差: 0.115657	校验误差:
half_half	实验	第 4 次	训练周期:3 300/391	训练误差: 0.115420	校验误差:
half_half	实验	第 4 次	训练周期:4 0/391	训练误差: 0.170076	校验误差:
half_half	实验	第 4 次	训练周期:4 100/391	训练误差: 0.104923	校验误差:
half_half	实验	第 4 次	训练周期:4 200/391	训练误差: 0.099128	校验误差:
half_half	实验	第 4 次	训练周期:4 300/391	训练误差: 0.102467	校验误差:
half_half	实验	第 4 次	训练周期:6 0/391	训练误差: 11.643244	校验误差:
half_half	实验	第 4 次	训练周期:6 100/391	训练误差: 1.368878	校验误差:
half_half	实验	第 4 次	训练周期:6 200/391	训练误差: 0.924077	校验误差:
half_half	实验	第 4 次	训练周期:6 300/391	训练误差: 0.731035	校验误差:
half_half	实验	第 4 次	训练周期:7 0/391	训练误差: 0.229996	校验误差:
half_half	实验	第 4 次	训练周期:7 100/391	训练误差: 0.280220	校验误差:
half_half	实验	第 4 次	训练周期:7 200/391	训练误差: 0.253336	校验误差:
half_half	实验	第 4 次	训练周期:7 300/391	训练误差: 0.239734	校验误差:
half_half	实验	第 4 次	训练周期:8 0/391	训练误差: 0.236984	校验误差:
half_half	实验	第 4 次	训练周期:8 100/391	训练误差: 0.192467	校验误差:
half_half	实验	第 4 次	训练周期:8 200/391	训练误差: 0.178721	校验误差:
half_half	实验	第 4 次	训练周期:8 300/391	训练误差: 0.177418	校验误差:
half_half	实验	第 4 次	训练周期:9 0/391	训练误差: 0.235965	校验误差:
half_half	实验	第 4 次	训练周期:9 100/391	训练误差: 0.150831	校验误差:
half_half	实验	第 4 次	训练周期:9 200/391	训练误差: 0.148433	校验误差:

In [113]: # 开始实验 2

```
experiment = 'half_all'
for time in range(times):
    net = ConvNet()
    criterion = nn.CrossEntropyLoss()
    optimizer = optim.SGD(net.parameters(), lr=0.001, momentum=0.9)
    record = []
    # 开始训练循环
    for epoch in range(int(num_epochs / 2)):
        losses = []
        train_rights = []
        for batch_idx, (data, target) in enumerate(train_loader1):
            data, target = Variable(data), Variable(target)
            net.train()
            output = net(data)
            loss = criterion(output, target)
            optimizer.zero_grad()
            loss.backward()
            optimizer.step()
            right = rightness(output, target)
            train_rights.append(right)
            losses.append(loss.data.numpy())

        if batch_idx % 100 == 0:
            train_r = 1.0 * np.sum([i[0] for i in train_rights]) / np.sum([i[1] for i in train_rights])
            net.eval() # 校验集上测试
            val_rights = []
            val_losses = []
            for (data, target) in validation_loader:
                data, target = Variable(data), Variable(target)
                output = net(data)
                right = rightness(output, target)
                val_rights.append(right)
                val_losses.append(loss.data.numpy())
            val_r = 1.0 * np.sum([i[0] for i in val_rights]) / np.sum([i[1] for i in val_rights])
            record.append([np.mean(losses), np.mean(val_losses), val_r])
```

```

        print('{}实验 \t第{}次 \t训练周期:{} {}/{} \t训练误差: {:.6f} \t校验误差:
              experiment, time, epoch, batch_idx, len(train_loader1),
              np.mean(losses), np.mean(val_losses), train_r, val_r))

for epoch in range(int(num_epochs/2 + 1), num_epochs):
    train_rights = []
    losses = []
    for batch_idx, (data, target) in enumerate(train_loader):
        data, target = Variable(data), Variable(target)
        net.train()
        output = net(data)
        loss = criterion(output, target)
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()
        right = rightness(output, target)
        train_rights.append(right)
        losses.append(loss.data.numpy())

    if batch_idx % 100 == 0:
        train_r = 1.0 * np.sum([i[0] for i in train_rights]) / np.sum([i[1] for i in
        net.eval()
        val_rights = []
        val_losses = []
        for (data, target) in validation_loader:
            data, target = Variable(data), Variable(target)
            output = net(data)
            right = rightness(output, target)
            val_rights.append(right)
            val_losses.append(loss.data.numpy())
        val_r = 1.0 * np.sum([i[0] for i in val_rights]) / np.sum([i[1] for i in
        record.append([np.mean(losses), np.mean(val_losses), val_r])
        print('{}实验 \t第{}次 \t训练周期:{} {}/{} \t训练误差: {:.6f} \t校验误差:
              experiment, time, epoch, batch_idx, len(train_loader),
              np.mean(losses), np.mean(val_losses), train_r, val_r))

net.eval() # 测试集上测试
test_rights = []
test_losses = []
for (data, target) in test_loader:

```

```

data, target = Variable(data), Variable(target)
output = net(data)
right = rightness(output, target)
test_rights.append(right)
test_losses.append(loss.data.numpy())
right_ratio = 1.0 * np.sum([i[0] for i in test_rights]) / np.sum([i[1] for i in test_rights])
results[(experiment,time)] = [record, right_ratio]

```

half_all	实验	第 0 次	训练周期:0 0/391	训练误差: 2.296002	校验误差:
half_all	实验	第 0 次	训练周期:0 100/391	训练误差: 2.142045	校验误差:
half_all	实验	第 0 次	训练周期:0 200/391	训练误差: 1.832251	校验误差:
half_all	实验	第 0 次	训练周期:0 300/391	训练误差: 1.481800	校验误差:
half_all	实验	第 0 次	训练周期:1 0/391	训练误差: 0.260728	校验误差:
half_all	实验	第 0 次	训练周期:1 100/391	训练误差: 0.226169	校验误差:
half_all	实验	第 0 次	训练周期:1 200/391	训练误差: 0.212184	校验误差:
half_all	实验	第 0 次	训练周期:1 300/391	训练误差: 0.201435	校验误差:
half_all	实验	第 0 次	训练周期:2 0/391	训练误差: 0.125058	校验误差:
half_all	实验	第 0 次	训练周期:2 100/391	训练误差: 0.148922	校验误差:
half_all	实验	第 0 次	训练周期:2 200/391	训练误差: 0.146683	校验误差:
half_all	实验	第 0 次	训练周期:2 300/391	训练误差: 0.145155	校验误差:
half_all	实验	第 0 次	训练周期:3 0/391	训练误差: 0.092340	校验误差:
half_all	实验	第 0 次	训练周期:3 100/391	训练误差: 0.129383	校验误差:
half_all	实验	第 0 次	训练周期:3 200/391	训练误差: 0.119221	校验误差:
half_all	实验	第 0 次	训练周期:3 300/391	训练误差: 0.114895	校验误差:
half_all	实验	第 0 次	训练周期:4 0/391	训练误差: 0.047596	校验误差:
half_all	实验	第 0 次	训练周期:4 100/391	训练误差: 0.099258	校验误差:
half_all	实验	第 0 次	训练周期:4 200/391	训练误差: 0.103699	校验误差:
half_all	实验	第 0 次	训练周期:4 300/391	训练误差: 0.100958	校验误差:
half_all	实验	第 0 次	训练周期:6 0/391	训练误差: 5.104378	校验误差:
half_all	实验	第 0 次	训练周期:6 100/391	训练误差: 1.128477	校验误差:
half_all	实验	第 0 次	训练周期:6 200/391	训练误差: 0.861930	校验误差:
half_all	实验	第 0 次	训练周期:6 300/391	训练误差: 0.736606	校验误差:
half_all	实验	第 0 次	训练周期:7 0/391	训练误差: 0.389897	校验误差:
half_all	实验	第 0 次	训练周期:7 100/391	训练误差: 0.395208	校验误差:
half_all	实验	第 0 次	训练周期:7 200/391	训练误差: 0.381831	校验误差:
half_all	实验	第 0 次	训练周期:7 300/391	训练误差: 0.364846	校验误差:
half_all	实验	第 0 次	训练周期:8 0/391	训练误差: 0.298304	校验误差:
half_all	实验	第 0 次	训练周期:8 100/391	训练误差: 0.315279	校验误差:
half_all	实验	第 0 次	训练周期:8 200/391	训练误差: 0.296174	校验误差:
half_all	实验	第 0 次	训练周期:8 300/391	训练误差: 0.285616	校验误差:

half_all	实验	第 0 次	训练周期:9 0/391	训练误差: 0.138789	校验误差:
half_all	实验	第 0 次	训练周期:9 100/391	训练误差: 0.252531	校验误差:
half_all	实验	第 0 次	训练周期:9 200/391	训练误差: 0.255277	校验误差:
half_all	实验	第 0 次	训练周期:9 300/391	训练误差: 0.249369	校验误差:
half_all	实验	第 1 次	训练周期:0 0/391	训练误差: 2.308880	校验误差:
half_all	实验	第 1 次	训练周期:0 100/391	训练误差: 2.084367	校验误差:
half_all	实验	第 1 次	训练周期:0 200/391	训练误差: 1.655064	校验误差:
half_all	实验	第 1 次	训练周期:0 300/391	训练误差: 1.255302	校验误差:
half_all	实验	第 1 次	训练周期:1 0/391	训练误差: 0.213647	校验误差:
half_all	实验	第 1 次	训练周期:1 100/391	训练误差: 0.206038	校验误差:
half_all	实验	第 1 次	训练周期:1 200/391	训练误差: 0.193070	校验误差:
half_all	实验	第 1 次	训练周期:1 300/391	训练误差: 0.183496	校验误差:
half_all	实验	第 1 次	训练周期:2 0/391	训练误差: 0.111266	校验误差:
half_all	实验	第 1 次	训练周期:2 100/391	训练误差: 0.152632	校验误差:
half_all	实验	第 1 次	训练周期:2 200/391	训练误差: 0.143065	校验误差:
half_all	实验	第 1 次	训练周期:2 300/391	训练误差: 0.138111	校验误差:
half_all	实验	第 1 次	训练周期:3 0/391	训练误差: 0.071359	校验误差:
half_all	实验	第 1 次	训练周期:3 100/391	训练误差: 0.113124	校验误差:
half_all	实验	第 1 次	训练周期:3 200/391	训练误差: 0.114660	校验误差:
half_all	实验	第 1 次	训练周期:3 300/391	训练误差: 0.110408	校验误差:
half_all	实验	第 1 次	训练周期:4 0/391	训练误差: 0.092776	校验误差:
half_all	实验	第 1 次	训练周期:4 100/391	训练误差: 0.095874	校验误差:
half_all	实验	第 1 次	训练周期:4 200/391	训练误差: 0.094602	校验误差:
half_all	实验	第 1 次	训练周期:4 300/391	训练误差: 0.093902	校验误差:
half_all	实验	第 1 次	训练周期:6 0/391	训练误差: 5.413808	校验误差:
half_all	实验	第 1 次	训练周期:6 100/391	训练误差: 1.150834	校验误差:
half_all	实验	第 1 次	训练周期:6 200/391	训练误差: 0.865080	校验误差:
half_all	实验	第 1 次	训练周期:6 300/391	训练误差: 0.734573	校验误差:
half_all	实验	第 1 次	训练周期:7 0/391	训练误差: 0.257880	校验误差:
half_all	实验	第 1 次	训练周期:7 100/391	训练误差: 0.377293	校验误差:
half_all	实验	第 1 次	训练周期:7 200/391	训练误差: 0.366380	校验误差:
half_all	实验	第 1 次	训练周期:7 300/391	训练误差: 0.358894	校验误差:
half_all	实验	第 1 次	训练周期:8 0/391	训练误差: 0.378638	校验误差:
half_all	实验	第 1 次	训练周期:8 100/391	训练误差: 0.314167	校验误差:
half_all	实验	第 1 次	训练周期:8 200/391	训练误差: 0.298686	校验误差:
half_all	实验	第 1 次	训练周期:8 300/391	训练误差: 0.294136	校验误差:
half_all	实验	第 1 次	训练周期:9 0/391	训练误差: 0.187008	校验误差:
half_all	实验	第 1 次	训练周期:9 100/391	训练误差: 0.255231	校验误差:
half_all	实验	第 1 次	训练周期:9 200/391	训练误差: 0.253317	校验误差:

half_all	实验	第 1 次	训练周期:9 300/391	训练误差: 0.249106	校验误差:
half_all	实验	第 2 次	训练周期:0 0/391	训练误差: 2.330858	校验误差:
half_all	实验	第 2 次	训练周期:0 100/391	训练误差: 2.130370	校验误差:
half_all	实验	第 2 次	训练周期:0 200/391	训练误差: 1.777254	校验误差:
half_all	实验	第 2 次	训练周期:0 300/391	训练误差: 1.387089	校验误差:
half_all	实验	第 2 次	训练周期:1 0/391	训练误差: 0.296862	校验误差:
half_all	实验	第 2 次	训练周期:1 100/391	训练误差: 0.215960	校验误差:
half_all	实验	第 2 次	训练周期:1 200/391	训练误差: 0.198089	校验误差:
half_all	实验	第 2 次	训练周期:1 300/391	训练误差: 0.191987	校验误差:
half_all	实验	第 2 次	训练周期:2 0/391	训练误差: 0.050062	校验误差:
half_all	实验	第 2 次	训练周期:2 100/391	训练误差: 0.150041	校验误差:
half_all	实验	第 2 次	训练周期:2 200/391	训练误差: 0.144346	校验误差:
half_all	实验	第 2 次	训练周期:2 300/391	训练误差: 0.140248	校验误差:
half_all	实验	第 2 次	训练周期:3 0/391	训练误差: 0.079446	校验误差:
half_all	实验	第 2 次	训练周期:3 100/391	训练误差: 0.115236	校验误差:
half_all	实验	第 2 次	训练周期:3 200/391	训练误差: 0.115434	校验误差:
half_all	实验	第 2 次	训练周期:3 300/391	训练误差: 0.113000	校验误差:
half_all	实验	第 2 次	训练周期:4 0/391	训练误差: 0.184583	校验误差:
half_all	实验	第 2 次	训练周期:4 100/391	训练误差: 0.108296	校验误差:
half_all	实验	第 2 次	训练周期:4 200/391	训练误差: 0.106678	校验误差:
half_all	实验	第 2 次	训练周期:4 300/391	训练误差: 0.101147	校验误差:
half_all	实验	第 2 次	训练周期:6 0/391	训练误差: 4.768227	校验误差:
half_all	实验	第 2 次	训练周期:6 100/391	训练误差: 1.108644	校验误差:
half_all	实验	第 2 次	训练周期:6 200/391	训练误差: 0.853416	校验误差:
half_all	实验	第 2 次	训练周期:6 300/391	训练误差: 0.732587	校验误差:
half_all	实验	第 2 次	训练周期:7 0/391	训练误差: 0.598022	校验误差:
half_all	实验	第 2 次	训练周期:7 100/391	训练误差: 0.396241	校验误差:
half_all	实验	第 2 次	训练周期:7 200/391	训练误差: 0.379480	校验误差:
half_all	实验	第 2 次	训练周期:7 300/391	训练误差: 0.367499	校验误差:
half_all	实验	第 2 次	训练周期:8 0/391	训练误差: 0.359207	校验误差:
half_all	实验	第 2 次	训练周期:8 100/391	训练误差: 0.299584	校验误差:
half_all	实验	第 2 次	训练周期:8 200/391	训练误差: 0.294662	校验误差:
half_all	实验	第 2 次	训练周期:8 300/391	训练误差: 0.292266	校验误差:
half_all	实验	第 2 次	训练周期:9 0/391	训练误差: 0.305194	校验误差:
half_all	实验	第 2 次	训练周期:9 100/391	训练误差: 0.258916	校验误差:
half_all	实验	第 2 次	训练周期:9 200/391	训练误差: 0.256152	校验误差:
half_all	实验	第 2 次	训练周期:9 300/391	训练误差: 0.248660	校验误差:
half_all	实验	第 3 次	训练周期:0 0/391	训练误差: 2.297813	校验误差:
half_all	实验	第 3 次	训练周期:0 100/391	训练误差: 2.107963	校验误差:

half_all	实验	第 3 次	训练周期:0 200/391	训练误差: 1.751329	校验误差:
half_all	实验	第 3 次	训练周期:0 300/391	训练误差: 1.369957	校验误差:
half_all	实验	第 3 次	训练周期:1 0/391	训练误差: 0.358036	校验误差:
half_all	实验	第 3 次	训练周期:1 100/391	训练误差: 0.236108	校验误差:
half_all	实验	第 3 次	训练周期:1 200/391	训练误差: 0.213373	校验误差:
half_all	实验	第 3 次	训练周期:1 300/391	训练误差: 0.194165	校验误差:
half_all	实验	第 3 次	训练周期:2 0/391	训练误差: 0.135449	校验误差:
half_all	实验	第 3 次	训练周期:2 100/391	训练误差: 0.149137	校验误差:
half_all	实验	第 3 次	训练周期:2 200/391	训练误差: 0.143014	校验误差:
half_all	实验	第 3 次	训练周期:2 300/391	训练误差: 0.135647	校验误差:
half_all	实验	第 3 次	训练周期:3 0/391	训练误差: 0.159445	校验误差:
half_all	实验	第 3 次	训练周期:3 100/391	训练误差: 0.108773	校验误差:
half_all	实验	第 3 次	训练周期:3 200/391	训练误差: 0.117075	校验误差:
half_all	实验	第 3 次	训练周期:3 300/391	训练误差: 0.114300	校验误差:
half_all	实验	第 3 次	训练周期:4 0/391	训练误差: 0.114838	校验误差:
half_all	实验	第 3 次	训练周期:4 100/391	训练误差: 0.103570	校验误差:
half_all	实验	第 3 次	训练周期:4 200/391	训练误差: 0.103636	校验误差:
half_all	实验	第 3 次	训练周期:4 300/391	训练误差: 0.101412	校验误差:
half_all	实验	第 3 次	训练周期:6 0/391	训练误差: 5.135493	校验误差:
half_all	实验	第 3 次	训练周期:6 100/391	训练误差: 1.148619	校验误差:
half_all	实验	第 3 次	训练周期:6 200/391	训练误差: 0.878931	校验误差:
half_all	实验	第 3 次	训练周期:6 300/391	训练误差: 0.746926	校验误差:
half_all	实验	第 3 次	训练周期:7 0/391	训练误差: 0.275325	校验误差:
half_all	实验	第 3 次	训练周期:7 100/391	训练误差: 0.384692	校验误差:
half_all	实验	第 3 次	训练周期:7 200/391	训练误差: 0.376896	校验误差:
half_all	实验	第 3 次	训练周期:7 300/391	训练误差: 0.363165	校验误差:
half_all	实验	第 3 次	训练周期:8 0/391	训练误差: 0.338732	校验误差:
half_all	实验	第 3 次	训练周期:8 100/391	训练误差: 0.302909	校验误差:
half_all	实验	第 3 次	训练周期:8 200/391	训练误差: 0.293084	校验误差:
half_all	实验	第 3 次	训练周期:8 300/391	训练误差: 0.290392	校验误差:
half_all	实验	第 3 次	训练周期:9 0/391	训练误差: 0.277083	校验误差:
half_all	实验	第 3 次	训练周期:9 100/391	训练误差: 0.268836	校验误差:
half_all	实验	第 3 次	训练周期:9 200/391	训练误差: 0.255757	校验误差:
half_all	实验	第 3 次	训练周期:9 300/391	训练误差: 0.250759	校验误差:
half_all	实验	第 4 次	训练周期:0 0/391	训练误差: 2.302861	校验误差:
half_all	实验	第 4 次	训练周期:0 100/391	训练误差: 2.147107	校验误差:
half_all	实验	第 4 次	训练周期:0 200/391	训练误差: 1.855389	校验误差:
half_all	实验	第 4 次	训练周期:0 300/391	训练误差: 1.546164	校验误差:
half_all	实验	第 4 次	训练周期:1 0/391	训练误差: 0.292606	校验误差:

half_all	实验	第 4 次	训练周期:1 100/391	训练误差: 0.243769	校验误差:
half_all	实验	第 4 次	训练周期:1 200/391	训练误差: 0.216647	校验误差:
half_all	实验	第 4 次	训练周期:1 300/391	训练误差: 0.203706	校验误差:
half_all	实验	第 4 次	训练周期:2 0/391	训练误差: 0.088193	校验误差:
half_all	实验	第 4 次	训练周期:2 100/391	训练误差: 0.151790	校验误差:
half_all	实验	第 4 次	训练周期:2 200/391	训练误差: 0.145822	校验误差:
half_all	实验	第 4 次	训练周期:2 300/391	训练误差: 0.139372	校验误差:
half_all	实验	第 4 次	训练周期:3 0/391	训练误差: 0.129110	校验误差:
half_all	实验	第 4 次	训练周期:3 100/391	训练误差: 0.115843	校验误差:
half_all	实验	第 4 次	训练周期:3 200/391	训练误差: 0.113639	校验误差:
half_all	实验	第 4 次	训练周期:3 300/391	训练误差: 0.115602	校验误差:
half_all	实验	第 4 次	训练周期:4 0/391	训练误差: 0.072636	校验误差:
half_all	实验	第 4 次	训练周期:4 100/391	训练误差: 0.097468	校验误差:
half_all	实验	第 4 次	训练周期:4 200/391	训练误差: 0.096697	校验误差:
half_all	实验	第 4 次	训练周期:4 300/391	训练误差: 0.097670	校验误差:
half_all	实验	第 4 次	训练周期:6 0/391	训练误差: 5.615283	校验误差:
half_all	实验	第 4 次	训练周期:6 100/391	训练误差: 1.150396	校验误差:
half_all	实验	第 4 次	训练周期:6 200/391	训练误差: 0.867902	校验误差:
half_all	实验	第 4 次	训练周期:6 300/391	训练误差: 0.734946	校验误差:
half_all	实验	第 4 次	训练周期:7 0/391	训练误差: 0.373266	校验误差:
half_all	实验	第 4 次	训练周期:7 100/391	训练误差: 0.392979	校验误差:
half_all	实验	第 4 次	训练周期:7 200/391	训练误差: 0.371083	校验误差:
half_all	实验	第 4 次	训练周期:7 300/391	训练误差: 0.355823	校验误差:
half_all	实验	第 4 次	训练周期:8 0/391	训练误差: 0.223790	校验误差:
half_all	实验	第 4 次	训练周期:8 100/391	训练误差: 0.290465	校验误差:
half_all	实验	第 4 次	训练周期:8 200/391	训练误差: 0.286311	校验误差:
half_all	实验	第 4 次	训练周期:8 300/391	训练误差: 0.282951	校验误差:
half_all	实验	第 4 次	训练周期:9 0/391	训练误差: 0.183038	校验误差:
half_all	实验	第 4 次	训练周期:9 100/391	训练误差: 0.248082	校验误差:
half_all	实验	第 4 次	训练周期:9 200/391	训练误差: 0.245263	校验误差:
half_all	实验	第 4 次	训练周期:9 300/391	训练误差: 0.242845	校验误差:

In [114]: # 开始实验 3

```

experiment = 'all'
for time in range(times):
    net = ConvNet()
    criterion = nn.CrossEntropyLoss()
    optimizer = optim.SGD(net.parameters(), lr=0.001, momentum=0.9)
    record = []

```

```

# 开始训练循环
for epoch in range(int(num_epochs)):
    losses = []
    train_rights = []
    for batch_idx, (data, target) in enumerate(train_loader):
        data, target = Variable(data), Variable(target)
        net.train()
        output = net(data)
        loss = criterion(output, target)
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()
        right = rightness(output, target)
        train_rights.append(right)
        losses.append(loss.data.numpy())

    if batch_idx % 100 == 0:
        train_r = 1.0 * np.sum([i[0] for i in train_rights]) / np.sum([i[1] for i in train_rights])
        net.eval()
        val_rights = []
        val_losses = []
        for (data, target) in validation_loader:
            data, target = Variable(data), Variable(target)
            output = net(data)
            right = rightness(output, target)
            val_rights.append(right)
            val_losses.append(loss.data.numpy())
        val_r = 1.0 * np.sum([i[0] for i in val_rights]) / np.sum([i[1] for i in val_rights])
        record.append([np.mean(losses), np.mean(val_losses), val_r])
        print('{}实验 \t第{}次 \t训练周期:{} {}/{} \t训练误差: {:.6f} \t校验误差: {:.6f}'.format(
            experiment, time, epoch, batch_idx, len(train_loader),
            np.mean(losses), np.mean(val_losses), train_r, val_r))
    net.eval()
    test_rights = []
    test_losses = []
    for (data, target) in test_loader:
        data, target = Variable(data), Variable(target)

```

```

output = net(data)
right = rightness(output, target)
test_rights.append(right)
test_losses.append(loss.data.numpy())
right_ratio = 1.0 * np.sum([i[0] for i in test_rights]) / np.sum([i[1] for i in test
results[(experiment,time)] = [record, right_ratio]

```

all 实验	第 0 次	训练周期:0 0/391	训练误差: 2.295899	校验误差: 2.295899
all 实验	第 0 次	训练周期:0 100/391	训练误差: 2.295106	校验误差: 2.295106
all 实验	第 0 次	训练周期:0 200/391	训练误差: 2.285955	校验误差: 2.285955
all 实验	第 0 次	训练周期:0 300/391	训练误差: 2.263304	校验误差: 2.263304
all 实验	第 0 次	训练周期:1 0/391	训练误差: 1.625435	校验误差: 1.625435
all 实验	第 0 次	训练周期:1 100/391	训练误差: 1.230956	校验误差: 0.984172
all 实验	第 0 次	训练周期:1 200/391	训练误差: 0.984172	校验误差: 0.625435
all 实验	第 0 次	训练周期:1 300/391	训练误差: 0.852158	校验误差: 0.625435
all 实验	第 0 次	训练周期:2 0/391	训练误差: 0.558360	校验误差: 0.558360
all 实验	第 0 次	训练周期:2 100/391	训练误差: 0.483009	校验误差: 0.483009
all 实验	第 0 次	训练周期:2 200/391	训练误差: 0.448554	校验误差: 0.448554
all 实验	第 0 次	训练周期:2 300/391	训练误差: 0.429144	校验误差: 0.429144
all 实验	第 0 次	训练周期:3 0/391	训练误差: 0.301850	校验误差: 0.301850
all 实验	第 0 次	训练周期:3 100/391	训练误差: 0.333253	校验误差: 0.333253
all 实验	第 0 次	训练周期:3 200/391	训练误差: 0.322547	校验误差: 0.322547
all 实验	第 0 次	训练周期:3 300/391	训练误差: 0.318453	校验误差: 0.318453
all 实验	第 0 次	训练周期:4 0/391	训练误差: 0.328627	校验误差: 0.328627
all 实验	第 0 次	训练周期:4 100/391	训练误差: 0.278920	校验误差: 0.278920
all 实验	第 0 次	训练周期:4 200/391	训练误差: 0.276911	校验误差: 0.276911
all 实验	第 0 次	训练周期:4 300/391	训练误差: 0.270551	校验误差: 0.270551
all 实验	第 0 次	训练周期:5 0/391	训练误差: 0.245349	校验误差: 0.245349
all 实验	第 0 次	训练周期:5 100/391	训练误差: 0.240023	校验误差: 0.240023
all 实验	第 0 次	训练周期:5 200/391	训练误差: 0.240829	校验误差: 0.240829
all 实验	第 0 次	训练周期:5 300/391	训练误差: 0.231520	校验误差: 0.231520
all 实验	第 0 次	训练周期:6 0/391	训练误差: 0.194556	校验误差: 0.194556
all 实验	第 0 次	训练周期:6 100/391	训练误差: 0.209834	校验误差: 0.209834
all 实验	第 0 次	训练周期:6 200/391	训练误差: 0.210146	校验误差: 0.210146
all 实验	第 0 次	训练周期:6 300/391	训练误差: 0.208293	校验误差: 0.208293
all 实验	第 0 次	训练周期:7 0/391	训练误差: 0.258905	校验误差: 0.258905
all 实验	第 0 次	训练周期:7 100/391	训练误差: 0.179332	校验误差: 0.179332
all 实验	第 0 次	训练周期:7 200/391	训练误差: 0.181011	校验误差: 0.181011
all 实验	第 0 次	训练周期:7 300/391	训练误差: 0.184565	校验误差: 0.184565
all 实验	第 0 次	训练周期:8 0/391	训练误差: 0.149967	校验误差: 0.149967

all 实验	第 0 次	训练周期:8 100/391	训练误差: 0.177115	校验误差: 0.1
all 实验	第 0 次	训练周期:8 200/391	训练误差: 0.168221	校验误差: 0.0
all 实验	第 0 次	训练周期:8 300/391	训练误差: 0.168649	校验误差: 0.2
all 实验	第 0 次	训练周期:9 0/391	训练误差: 0.140992	校验误差: 0.14
all 实验	第 0 次	训练周期:9 100/391	训练误差: 0.160917	校验误差: 0.2
all 实验	第 0 次	训练周期:9 200/391	训练误差: 0.160727	校验误差: 0.1
all 实验	第 0 次	训练周期:9 300/391	训练误差: 0.158102	校验误差: 0.0
all 实验	第 1 次	训练周期:0 0/391	训练误差: 2.302556	校验误差: 2.302
all 实验	第 1 次	训练周期:0 100/391	训练误差: 2.298899	校验误差: 2.2
all 实验	第 1 次	训练周期:0 200/391	训练误差: 2.290584	校验误差: 2.2
all 实验	第 1 次	训练周期:0 300/391	训练误差: 2.271713	校验误差: 2.2
all 实验	第 1 次	训练周期:1 0/391	训练误差: 1.835304	校验误差: 1.835
all 实验	第 1 次	训练周期:1 100/391	训练误差: 1.276877	校验误差: 0.9
all 实验	第 1 次	训练周期:1 200/391	训练误差: 1.020495	校验误差: 0.8
all 实验	第 1 次	训练周期:1 300/391	训练误差: 0.887245	校验误差: 0.5
all 实验	第 1 次	训练周期:2 0/391	训练误差: 0.472775	校验误差: 0.472
all 实验	第 1 次	训练周期:2 100/391	训练误差: 0.463955	校验误差: 0.4
all 实验	第 1 次	训练周期:2 200/391	训练误差: 0.438126	校验误差: 0.5
all 实验	第 1 次	训练周期:2 300/391	训练误差: 0.419175	校验误差: 0.3
all 实验	第 1 次	训练周期:3 0/391	训练误差: 0.293047	校验误差: 0.293
all 实验	第 1 次	训练周期:3 100/391	训练误差: 0.337284	校验误差: 0.3
all 实验	第 1 次	训练周期:3 200/391	训练误差: 0.343112	校验误差: 0.4
all 实验	第 1 次	训练周期:3 300/391	训练误差: 0.328177	校验误差: 0.2
all 实验	第 1 次	训练周期:4 0/391	训练误差: 0.248867	校验误差: 0.248
all 实验	第 1 次	训练周期:4 100/391	训练误差: 0.255237	校验误差: 0.2
all 实验	第 1 次	训练周期:4 200/391	训练误差: 0.264296	校验误差: 0.1
all 实验	第 1 次	训练周期:4 300/391	训练误差: 0.267187	校验误差: 0.1
all 实验	第 1 次	训练周期:5 0/391	训练误差: 0.136257	校验误差: 0.136
all 实验	第 1 次	训练周期:5 100/391	训练误差: 0.237257	校验误差: 0.2
all 实验	第 1 次	训练周期:5 200/391	训练误差: 0.230644	校验误差: 0.2
all 实验	第 1 次	训练周期:5 300/391	训练误差: 0.226397	校验误差: 0.1
all 实验	第 1 次	训练周期:6 0/391	训练误差: 0.180649	校验误差: 0.180
all 实验	第 1 次	训练周期:6 100/391	训练误差: 0.199367	校验误差: 0.3
all 实验	第 1 次	训练周期:6 200/391	训练误差: 0.201634	校验误差: 0.2
all 实验	第 1 次	训练周期:6 300/391	训练误差: 0.197238	校验误差: 0.1
all 实验	第 1 次	训练周期:7 0/391	训练误差: 0.260119	校验误差: 0.260
all 实验	第 1 次	训练周期:7 100/391	训练误差: 0.188797	校验误差: 0.1
all 实验	第 1 次	训练周期:7 200/391	训练误差: 0.185931	校验误差: 0.1
all 实验	第 1 次	训练周期:7 300/391	训练误差: 0.182475	校验误差: 0.2

all 实验	第 1 次	训练周期:8 0/391	训练误差: 0.107106	校验误差: 0.107106
all 实验	第 1 次	训练周期:8 100/391	训练误差: 0.146183	校验误差: 0.146183
all 实验	第 1 次	训练周期:8 200/391	训练误差: 0.158864	校验误差: 0.158864
all 实验	第 1 次	训练周期:8 300/391	训练误差: 0.165504	校验误差: 0.165504
all 实验	第 1 次	训练周期:9 0/391	训练误差: 0.141614	校验误差: 0.141614
all 实验	第 1 次	训练周期:9 100/391	训练误差: 0.156238	校验误差: 0.156238
all 实验	第 1 次	训练周期:9 200/391	训练误差: 0.149625	校验误差: 0.149625
all 实验	第 1 次	训练周期:9 300/391	训练误差: 0.148266	校验误差: 0.148266
all 实验	第 2 次	训练周期:0 0/391	训练误差: 2.304732	校验误差: 2.304732
all 实验	第 2 次	训练周期:0 100/391	训练误差: 2.302661	校验误差: 2.302661
all 实验	第 2 次	训练周期:0 200/391	训练误差: 2.298404	校验误差: 2.298404
all 实验	第 2 次	训练周期:0 300/391	训练误差: 2.294719	校验误差: 2.294719
all 实验	第 2 次	训练周期:1 0/391	训练误差: 2.263099	校验误差: 2.263099
all 实验	第 2 次	训练周期:1 100/391	训练误差: 2.243481	校验误差: 2.243481
all 实验	第 2 次	训练周期:1 200/391	训练误差: 2.186789	校验误差: 2.186789
all 实验	第 2 次	训练周期:1 300/391	训练误差: 2.013180	校验误差: 2.013180
all 实验	第 2 次	训练周期:2 0/391	训练误差: 0.768967	校验误差: 0.768967
all 实验	第 2 次	训练周期:2 100/391	训练误差: 0.698144	校验误差: 0.698144
all 实验	第 2 次	训练周期:2 200/391	训练误差: 0.635792	校验误差: 0.635792
all 实验	第 2 次	训练周期:2 300/391	训练误差: 0.579836	校验误差: 0.579836
all 实验	第 2 次	训练周期:3 0/391	训练误差: 0.360257	校验误差: 0.360257
all 实验	第 2 次	训练周期:3 100/391	训练误差: 0.409968	校验误差: 0.409968
all 实验	第 2 次	训练周期:3 200/391	训练误差: 0.394541	校验误差: 0.394541
all 实验	第 2 次	训练周期:3 300/391	训练误差: 0.378982	校验误差: 0.378982
all 实验	第 2 次	训练周期:4 0/391	训练误差: 0.383539	校验误差: 0.383539
all 实验	第 2 次	训练周期:4 100/391	训练误差: 0.308560	校验误差: 0.308560
all 实验	第 2 次	训练周期:4 200/391	训练误差: 0.296614	校验误差: 0.296614
all 实验	第 2 次	训练周期:4 300/391	训练误差: 0.288393	校验误差: 0.288393
all 实验	第 2 次	训练周期:5 0/391	训练误差: 0.209082	校验误差: 0.209082
all 实验	第 2 次	训练周期:5 100/391	训练误差: 0.262217	校验误差: 0.262217
all 实验	第 2 次	训练周期:5 200/391	训练误差: 0.257396	校验误差: 0.257396
all 实验	第 2 次	训练周期:5 300/391	训练误差: 0.243809	校验误差: 0.243809
all 实验	第 2 次	训练周期:6 0/391	训练误差: 0.309433	校验误差: 0.309433
all 实验	第 2 次	训练周期:6 100/391	训练误差: 0.211038	校验误差: 0.211038
all 实验	第 2 次	训练周期:6 200/391	训练误差: 0.219357	校验误差: 0.219357
all 实验	第 2 次	训练周期:6 300/391	训练误差: 0.210314	校验误差: 0.210314
all 实验	第 2 次	训练周期:7 0/391	训练误差: 0.087516	校验误差: 0.087516
all 实验	第 2 次	训练周期:7 100/391	训练误差: 0.185726	校验误差: 0.185726
all 实验	第 2 次	训练周期:7 200/391	训练误差: 0.191244	校验误差: 0.191244

all 实验	第 2 次	训练周期:7 300/391	训练误差: 0.187879	校验误差: 0.3
all 实验	第 2 次	训练周期:8 0/391	训练误差: 0.211559	校验误差: 0.211
all 实验	第 2 次	训练周期:8 100/391	训练误差: 0.177323	校验误差: 0.1
all 实验	第 2 次	训练周期:8 200/391	训练误差: 0.174233	校验误差: 0.0
all 实验	第 2 次	训练周期:8 300/391	训练误差: 0.168022	校验误差: 0.1
all 实验	第 2 次	训练周期:9 0/391	训练误差: 0.152421	校验误差: 0.152
all 实验	第 2 次	训练周期:9 100/391	训练误差: 0.152478	校验误差: 0.1
all 实验	第 2 次	训练周期:9 200/391	训练误差: 0.145564	校验误差: 0.1
all 实验	第 2 次	训练周期:9 300/391	训练误差: 0.148166	校验误差: 0.0
all 实验	第 3 次	训练周期:0 0/391	训练误差: 2.308254	校验误差: 2.308
all 实验	第 3 次	训练周期:0 100/391	训练误差: 2.299373	校验误差: 2.2
all 实验	第 3 次	训练周期:0 200/391	训练误差: 2.295859	校验误差: 2.2
all 实验	第 3 次	训练周期:0 300/391	训练误差: 2.289448	校验误差: 2.2
all 实验	第 3 次	训练周期:1 0/391	训练误差: 2.194024	校验误差: 2.194
all 实验	第 3 次	训练周期:1 100/391	训练误差: 2.020583	校验误差: 1.8
all 实验	第 3 次	训练周期:1 200/391	训练误差: 1.637472	校验误差: 0.7
all 实验	第 3 次	训练周期:1 300/391	训练误差: 1.356849	校验误差: 0.7
all 实验	第 3 次	训练周期:2 0/391	训练误差: 0.698904	校验误差: 0.698
all 实验	第 3 次	训练周期:2 100/391	训练误差: 0.542018	校验误差: 0.3
all 实验	第 3 次	训练周期:2 200/391	训练误差: 0.522459	校验误差: 0.3
all 实验	第 3 次	训练周期:2 300/391	训练误差: 0.501618	校验误差: 0.5
all 实验	第 3 次	训练周期:3 0/391	训练误差: 0.398475	校验误差: 0.398
all 实验	第 3 次	训练周期:3 100/391	训练误差: 0.387839	校验误差: 0.3
all 实验	第 3 次	训练周期:3 200/391	训练误差: 0.373660	校验误差: 0.5
all 实验	第 3 次	训练周期:3 300/391	训练误差: 0.363862	校验误差: 0.3
all 实验	第 3 次	训练周期:4 0/391	训练误差: 0.272878	校验误差: 0.272
all 实验	第 3 次	训练周期:4 100/391	训练误差: 0.324473	校验误差: 0.4
all 实验	第 3 次	训练周期:4 200/391	训练误差: 0.306716	校验误差: 0.1
all 实验	第 3 次	训练周期:4 300/391	训练误差: 0.299427	校验误差: 0.3
all 实验	第 3 次	训练周期:5 0/391	训练误差: 0.308827	校验误差: 0.308
all 实验	第 3 次	训练周期:5 100/391	训练误差: 0.276334	校验误差: 0.1
all 实验	第 3 次	训练周期:5 200/391	训练误差: 0.260010	校验误差: 0.3
all 实验	第 3 次	训练周期:5 300/391	训练误差: 0.255091	校验误差: 0.4
all 实验	第 3 次	训练周期:6 0/391	训练误差: 0.277707	校验误差: 0.277
all 实验	第 3 次	训练周期:6 100/391	训练误差: 0.226570	校验误差: 0.3
all 实验	第 3 次	训练周期:6 200/391	训练误差: 0.227291	校验误差: 0.4
all 实验	第 3 次	训练周期:6 300/391	训练误差: 0.223993	校验误差: 0.2
all 实验	第 3 次	训练周期:7 0/391	训练误差: 0.273304	校验误差: 0.273
all 实验	第 3 次	训练周期:7 100/391	训练误差: 0.202415	校验误差: 0.2

all 实验	第 3 次	训练周期:7 200/391	训练误差: 0.200703	校验误差: 0.3
all 实验	第 3 次	训练周期:7 300/391	训练误差: 0.200168	校验误差: 0.2
all 实验	第 3 次	训练周期:8 0/391	训练误差: 0.044616	校验误差: 0.044
all 实验	第 3 次	训练周期:8 100/391	训练误差: 0.179395	校验误差: 0.1
all 实验	第 3 次	训练周期:8 200/391	训练误差: 0.183577	校验误差: 0.2
all 实验	第 3 次	训练周期:8 300/391	训练误差: 0.184103	校验误差: 0.0
all 实验	第 3 次	训练周期:9 0/391	训练误差: 0.178321	校验误差: 0.178
all 实验	第 3 次	训练周期:9 100/391	训练误差: 0.178771	校验误差: 0.1
all 实验	第 3 次	训练周期:9 200/391	训练误差: 0.175687	校验误差: 0.1
all 实验	第 3 次	训练周期:9 300/391	训练误差: 0.173485	校验误差: 0.1
all 实验	第 4 次	训练周期:0 0/391	训练误差: 2.303575	校验误差: 2.303
all 实验	第 4 次	训练周期:0 100/391	训练误差: 2.298301	校验误差: 2.2
all 实验	第 4 次	训练周期:0 200/391	训练误差: 2.288746	校验误差: 2.2
all 实验	第 4 次	训练周期:0 300/391	训练误差: 2.268519	校验误差: 2.1
all 实验	第 4 次	训练周期:1 0/391	训练误差: 1.845823	校验误差: 1.845
all 实验	第 4 次	训练周期:1 100/391	训练误差: 1.256069	校验误差: 0.8
all 实验	第 4 次	训练周期:1 200/391	训练误差: 0.997480	校验误差: 0.5
all 实验	第 4 次	训练周期:1 300/391	训练误差: 0.854708	校验误差: 0.6
all 实验	第 4 次	训练周期:2 0/391	训练误差: 0.473887	校验误差: 0.473
all 实验	第 4 次	训练周期:2 100/391	训练误差: 0.429587	校验误差: 0.3
all 实验	第 4 次	训练周期:2 200/391	训练误差: 0.417110	校验误差: 0.4
all 实验	第 4 次	训练周期:2 300/391	训练误差: 0.404035	校验误差: 0.4
all 实验	第 4 次	训练周期:3 0/391	训练误差: 0.460981	校验误差: 0.460
all 实验	第 4 次	训练周期:3 100/391	训练误差: 0.313756	校验误差: 0.3
all 实验	第 4 次	训练周期:3 200/391	训练误差: 0.305941	校验误差: 0.3
all 实验	第 4 次	训练周期:3 300/391	训练误差: 0.299896	校验误差: 0.1
all 实验	第 4 次	训练周期:4 0/391	训练误差: 0.242440	校验误差: 0.242
all 实验	第 4 次	训练周期:4 100/391	训练误差: 0.252031	校验误差: 0.2
all 实验	第 4 次	训练周期:4 200/391	训练误差: 0.249224	校验误差: 0.2
all 实验	第 4 次	训练周期:4 300/391	训练误差: 0.246029	校验误差: 0.1
all 实验	第 4 次	训练周期:5 0/391	训练误差: 0.228573	校验误差: 0.228
all 实验	第 4 次	训练周期:5 100/391	训练误差: 0.221046	校验误差: 0.3
all 实验	第 4 次	训练周期:5 200/391	训练误差: 0.215197	校验误差: 0.2
all 实验	第 4 次	训练周期:5 300/391	训练误差: 0.211196	校验误差: 0.3
all 实验	第 4 次	训练周期:6 0/391	训练误差: 0.140006	校验误差: 0.140
all 实验	第 4 次	训练周期:6 100/391	训练误差: 0.193925	校验误差: 0.1
all 实验	第 4 次	训练周期:6 200/391	训练误差: 0.190913	校验误差: 0.1
all 实验	第 4 次	训练周期:6 300/391	训练误差: 0.186329	校验误差: 0.1
all 实验	第 4 次	训练周期:7 0/391	训练误差: 0.132169	校验误差: 0.132

all 实验	第 4 次	训练周期:7 100/391	训练误差: 0.162737	校验误差: 0.0
all 实验	第 4 次	训练周期:7 200/391	训练误差: 0.165669	校验误差: 0.2
all 实验	第 4 次	训练周期:7 300/391	训练误差: 0.161550	校验误差: 0.1
all 实验	第 4 次	训练周期:8 0/391	训练误差: 0.141805	校验误差: 0.14
all 实验	第 4 次	训练周期:8 100/391	训练误差: 0.155245	校验误差: 0.0
all 实验	第 4 次	训练周期:8 200/391	训练误差: 0.154635	校验误差: 0.1
all 实验	第 4 次	训练周期:8 300/391	训练误差: 0.153564	校验误差: 0.0
all 实验	第 4 次	训练周期:9 0/391	训练误差: 0.191748	校验误差: 0.19
all 实验	第 4 次	训练周期:9 100/391	训练误差: 0.152378	校验误差: 0.1
all 实验	第 4 次	训练周期:9 200/391	训练误差: 0.148047	校验误差: 0.0
all 实验	第 4 次	训练周期:9 300/391	训练误差: 0.143195	校验误差: 0.1

0.2 实验结果

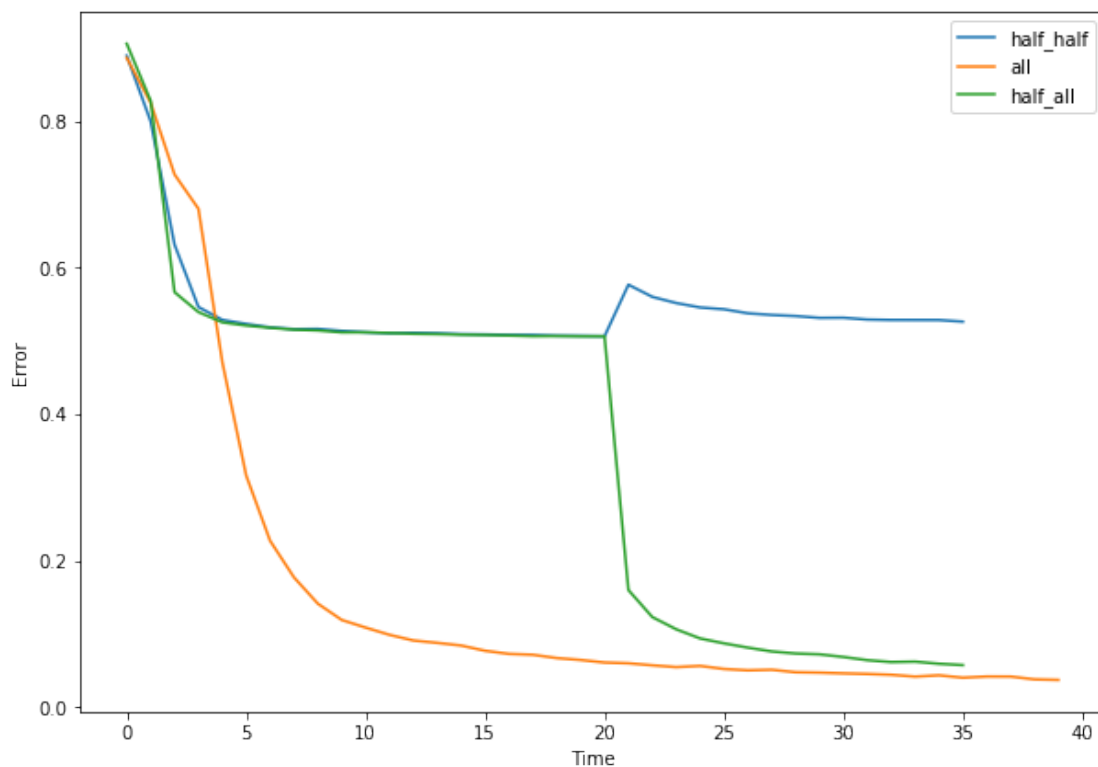
分别绘制校验集错误率曲线, 训练集和校验集 Loss 曲线, 并对比三种情况下测试集上分类准确度

In [115]: # 绘制校验集测试错误率曲线

```
one_curve = {}
tests = {}
for experiment in ['all', 'half_all', 'half_half']:
    one_experiment = []
    test_value = []
    for time in range(times):
        rr = results[(experiment, time)]
        one_experiment.append([ii[2] for ii in rr[0]])
        test_value.append(rr[1])
    aa = np.array(one_experiment)
    one_curve[(experiment)] = np.mean(aa, 0)
    tests[(experiment)] = np.mean(test_value)
```

```
In [117]: plt.figure(figsize = (10, 7))
plt.plot(1 - one_curve[('half_half')], label = 'half_half')
plt.plot(1 - one_curve[('all')], label = 'all')
plt.plot(1 - one_curve[('half_all')], label = 'half_all')
plt.legend()
plt.xlabel('Time')
plt.ylabel('Error')
```

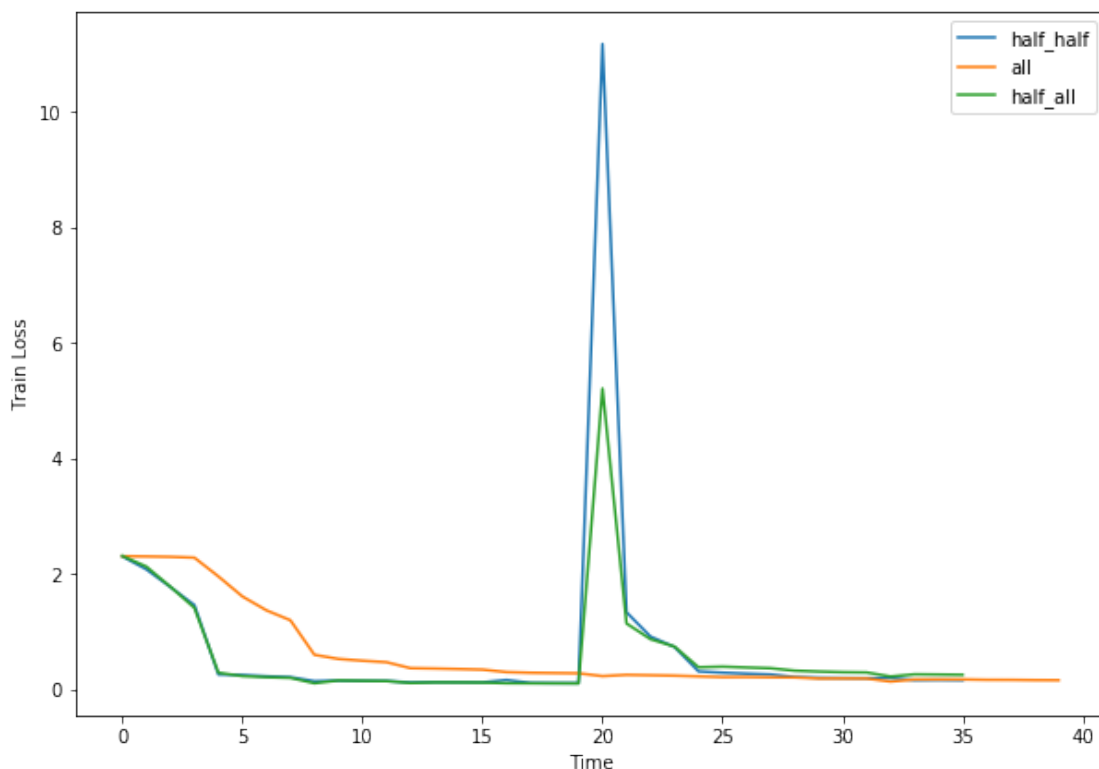
Out[117]: <matplotlib.text.Text at 0x2b1817799b0>



In [118]: # 绘制训练集 *loss*

```
one_curve = {}
for experiment in ['all', 'half_all', 'half_half']:
    one_experiment = []
    for time in range(times):
        rr = results[(experiment, time)]
        one_experiment.append([ii[0] for ii in rr[0]])
    aa = np.array(one_experiment)
    one_curve[(experiment)] = np.mean(aa, 0)
plt.figure(figsize = (10, 7))
plt.plot(one_curve[('half_half')], label = 'half_half')
plt.plot(one_curve[('all')], label = 'all')
plt.plot(one_curve[('half_all')], label = 'half_all')
plt.legend()
plt.xlabel('Time')
plt.ylabel('Train Loss')
```

Out[118]: <matplotlib.text.Text at 0x2b1814f1cc0>



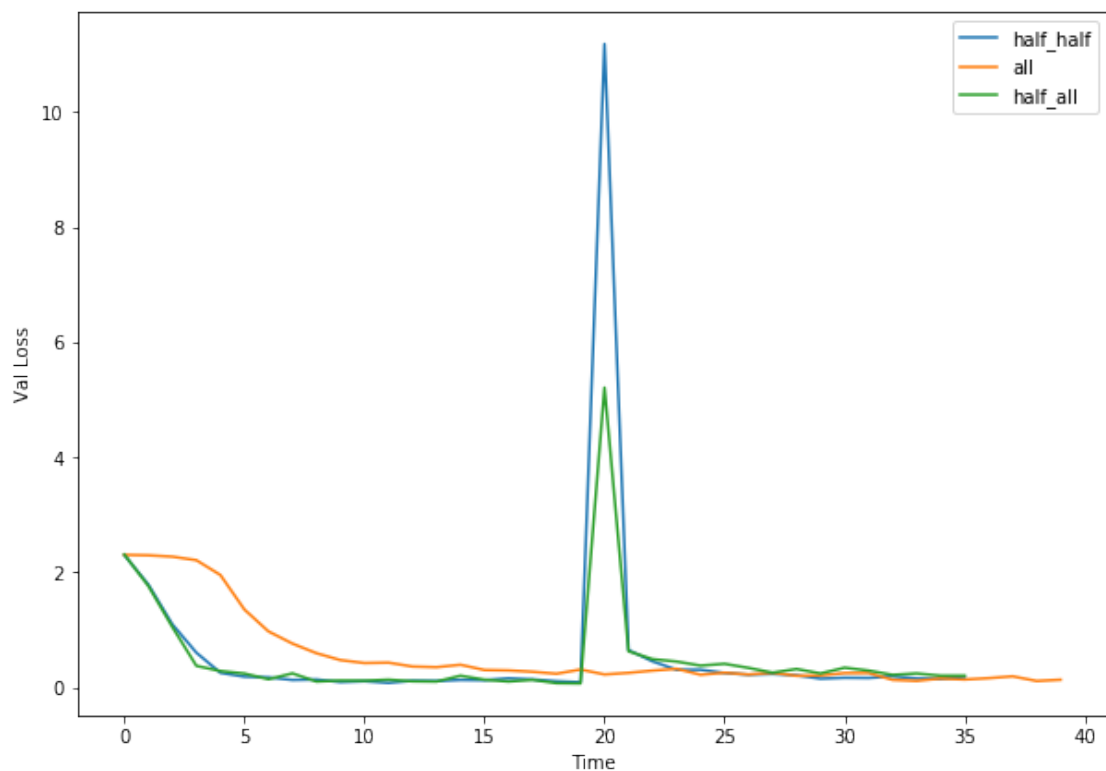
In [119]: # 绘制校验集 loss

```

one_curve = {}
for experiment in ['all', 'half_all', 'half_half']:
    one_experiment = []
    for time in range(times):
        rr = results[(experiment, time)]
        one_experiment.append([ii[1] for ii in rr[0]])
    aa = np.array(one_experiment)
    one_curve[(experiment)] = np.mean(aa, 0)
plt.figure(figsize = (10, 7))
plt.plot(one_curve[('half_half')], label = 'half_half')
plt.plot(one_curve[('all')], label = 'all')
plt.plot(one_curve[('half_all')], label = 'half_all')
plt.legend()
plt.xlabel('Time')
plt.ylabel('Val Loss')

```

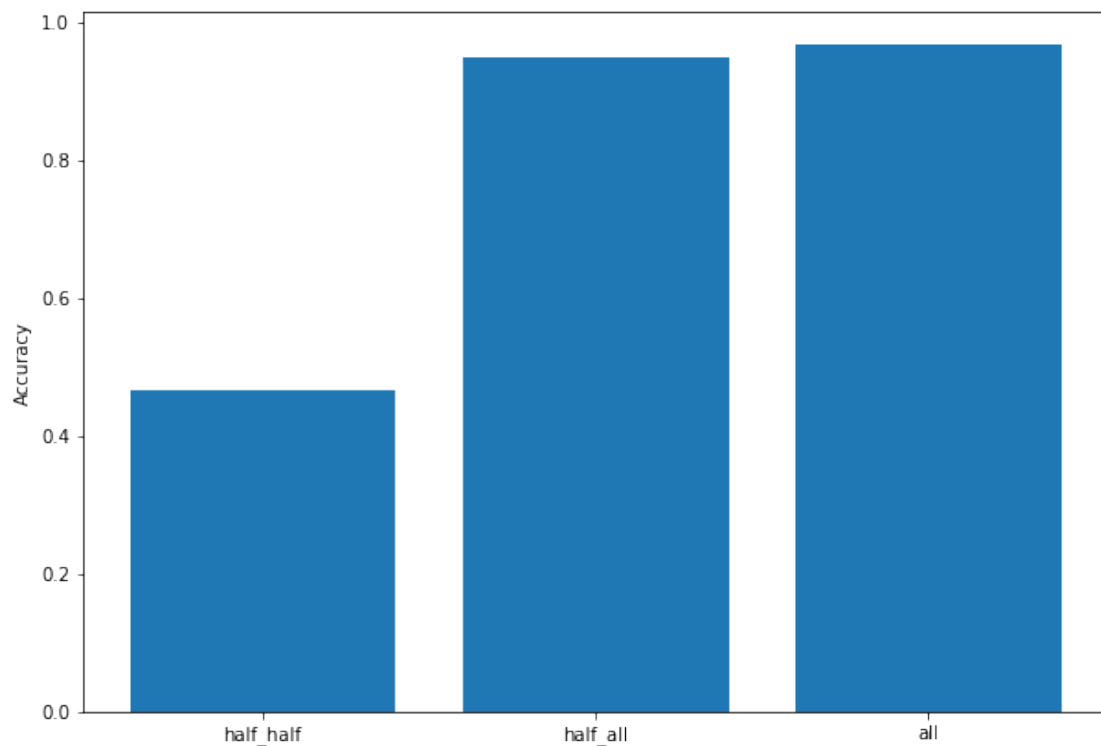
Out[119]: <matplotlib.text.Text at 0x2b180645748>



In [129]: # 绘制三种情况下测试集下准确率柱状图

```
accuracy = []
labels = ['half_half', 'half_all', 'all']
for experiment in labels:
    accuracy.append(tests[experiment])
plt.figure(figsize = (10, 7))
plt.bar(range(len(accuracy)), accuracy, tick_label=labels)
plt.ylabel('Accuracy')
```

Out[129]: <matplotlib.text.Text at 0x2b181aafda0>



0.3 结论

1. 采用 0-4 训练后，此时采用 0-9 训练校验集错误率下降比直接用 0-9 训练要快，但测试集准确率要小于直接用 0-9 在整个 epoch 训练
2. 采用 0-4 训练后，采用 5-9 训练的得到的网络分类准确率只有不到一半