In [ ]:

```
# Solution 2.3 Test):

import torch
import torch.nn as nn
import torch.optim as optim
import torchvision
import torch.utils.data as data
from torchvision import datasets, transforms
import matplotlib.pyplot as plt
import numpy as np
import copy
import math
import traceback

torch.cuda.empty_cache()
# Define the transformations to be applied to the images
transform = transforms.Compose([
    transforms.ToTensor(),
    transforms.Normalize((0.5,), (0.5,)),
    transforms.Resize((32,32))
])

# Load the fashionMNIST dataset
train_set = datasets.FashionMNIST('./data', train=True, download=True, transform=transform)

# ConvModule: a convolutional module in the above picture, consists a 2d convolutional layer, a 2d batchnorm layer, and a ReLU activation.
class ConvModule(nn.Module):
    def __init__(self, in_channels: int, out_channels: int, kernel_size, stride, padding='same'):
        super(ConvModule, self).__init__()
        self.conv2d = nn.Conv2d(
            in_channels, out_channels, kernel_size, stride=stride, padding=padding)

        self.batchnorm = nn.BatchNorm2d(out_channels)
        self.relu = nn.ReLU()

    def forward(self, x):
        x = self.conv2d(x)
        x = self.batchnorm(x)
        x = self.relu(x)

        return x

# InceptionModule: a inception module in the above picture, consists a convolution module with 1x1 filter,
# a convolution module with 3x3 filter, then concatenate these two outputs.
class InceptionModule(nn.Module):
    def __init__(self, in_channels, ch1x1, ch3x3):
        super(InceptionModule, self).__init__()

        self.conv1x1 = ConvModule(in_channels, ch1x1, (1, 1), 1)
        self.conv3x3 = ConvModule(in_channels, ch3x3, (3, 3), 1)

    def forward(self, x):
        out1 = self.conv1x1(x)
        out2 = self.conv3x3(x)
        x = torch.cat((out1, out2), 1)
        return x

# DownsampleModule: a downsample module in the above picture, consists a convolution module with 3x3 filter,
# a 2d maxpool layer, then concatenate these two outputs.
class DownsampleModule(nn.Module):
    def __init__(self, in_channels, out_channels):
        super(DownsampleModule, self).__init__()

        self.conv3x3 = ConvModule(in_channels, out_channels, (3, 3), (2, 2), padding='valid')
        self.maxpool = nn.MaxPool2d(kernel_size=3, stride=2)

    def forward(self, x):
        out1 = self.conv3x3(x)
        out2 = self.maxpool(x)

        #return out1
        x = torch.cat((out1, out2), 1)

        return x


# MiniGoogLeNet: the MiniGoogLeNet model. Input: input_channels * 32 * 32.
# When input_channels is 1, the input is a grayscale image. When input_channels is 3, the input is a RGB image.
# Output: a tensor with the shape of [-1, classes], where classes it the number of classes.

class MiniGoogLeNet(nn.Module):
    def __init__(self, classes, input_channels):
        super(MiniGoogLeNet, self).__init__()

        self.conv1 = ConvModule(input_channels, 96, kernel_size=(3, 3), stride=1) # input_channel is 3 if you want to deal with RGB image, 1 for grey scale image
        self.inception1 = InceptionModule(96, 32, 32)
        self.inception2 = InceptionModule(32+32, 32, 48)
        self.downsample1 = DownsampleModule(32+48, 80)

        self.inception3 = InceptionModule(80+80, 112, 48)
        self.inception4 = InceptionModule(112+48, 96, 64)
        self.inception5 = InceptionModule(96+64, 80, 80)
        self.inception6 = InceptionModule(80+80, 48, 96)
        self.downsample2 = DownsampleModule(48+96, 96)

        self.inception7 = InceptionModule(96+96, 176, 160)
        self.inception8 = InceptionModule(176+160, 176, 160)
        self.avgpool2d = nn.AvgPool2d(kernel_size=7)
        self.dropout = nn.Dropout2d(0.5)

        self.fc = nn.Linear(240, classes)
        #self.softmax = nn.Softmax(dim=-1)

    def forward(self, x):
        x = self.conv1(x)
        #print(x.shape)
        x = self.inception1(x)
        x = self.inception2(x)
        x = self.downsample1(x)

        x = self.inception3(x)
        x = self.inception4(x)
        x = self.inception5(x)
        x = self.inception6(x)
        x = self.downsample2(x)

        x = self.avgpool2d(x)
        x = self.dropout(x)

        x = torch.flatten(x, 1)
        x = self.fc(x)
        #x = self.softmax(x), no need for softmax because PyTorch Cross Entropy Loss implemented softmax

        return x

lr = 1e-09
BatchSize = [32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384]
# BatchSize = [32, 64, 128, 256, 512, 1024, 2048]
VALID_RATIO = 0.9
batch_size_to_loss = {}

def calculate_accuracy(y_pred,y):
    top_pred=y_pred.argmax(1,keepdim= True)
    correct =top_pred.eq(y.view_as(top_pred)).sum()
    acc=correct.float()/y.shape[0]
    return acc

def train(model, iterator, optimizer, criterion, device):
    epoch_loss = 0
    epoch_acc = 0
    model.train()
    for (x, y) in iterator:
        x = x.to(device)
        y = y.to(device)
        optimizer.zero_grad()
        y_pred = model(x)
        loss = criterion(y_pred, y)
        acc = calculate_accuracy(y_pred, y)
        loss.backward()
        optimizer.step()
        epoch_loss += loss.item()
        epoch_acc += acc.item()
    return epoch_loss / len(iterator), epoch_acc / len(iterator)

def evaluate(model, iterator, criterion, device):
    epoch_loss = 0
    epoch_acc = 0
    model.eval()
    with torch.no_grad():
        for (x, y) in iterator:
            x = x.to(device)
            y = y.to(device)
            y_pred= model(x)
            loss = criterion(y_pred, y)
            acc = calculate_accuracy(y_pred, y)
            epoch_loss += loss.item()
            epoch_acc += acc.item()
    return epoch_loss / len(iterator), epoch_acc / len(iterator)

device = torch.device("cuda:0" if torch.cuda.is_available() else "cpu")
for batchSize in BatchSize:
    torch.cuda.empty_cache()
    print(f"Batch Size: {batchSize}")
    torch.cuda.empty_cache()
    transformations = transforms.Compose([transforms.Resize((32, 32)),
                                transforms.ToTensor(),
                               ])
    training = datasets.FashionMNIST(root='./data', train=True, download=True, transform=transformations)
    n_train_examples = int(len(training) * VALID_RATIO)
    n_valid_examples = len(training) - n_train_examples
    t, v = data.random_split(training, [n_train_examples, n_valid_examples])
    v = copy.deepcopy(v)
    train_set = torch.utils.data.DataLoader(t, batch_size=batchSize, shuffle=True)
    valid_set = torch.utils.data.DataLoader(v, batch_size=batchSize, shuffle=True)
    model = MiniGoogLeNet(classes=10, input_channels=1).to(device)
    criterion = nn.CrossEntropyLoss()
    optimizer = optim.SGD(model.parameters(), lr=lr, momentum=0.9)
    best_loss=float('inf')

    for epoch in range(5):
      print("Current ecoch: ", epoch)
      train_loss,train_acc = train(model,train_set,optimizer,criterion,device)
      val_loss,val_acc = evaluate(model,valid_set,criterion,device)
      print(f'\tTrain Loss: {train_loss:.3f} | Train Acc: {train_acc*100:.2f}%')
      print(f'\tValidation Loss: {val_loss:.3f} | Validation Acc: {val_acc*100:.2f}%')
      if val_loss < best_loss:
        best_loss=val_loss
    batch_size_to_loss[batchSize] = best_loss
```

```
Batch Size: 32
Current ecoch:  0
        Train Loss: 0.644 | Train Acc: 76.40%
        Validation Loss: 0.558 | Validation Acc: 81.04%
Current ecoch:  1
        Train Loss: 0.388 | Train Acc: 85.91%
        Validation Loss: 0.329 | Validation Acc: 87.84%
Current ecoch:  2
        Train Loss: 0.322 | Train Acc: 88.59%
        Validation Loss: 0.308 | Validation Acc: 89.12%
Current ecoch:  3
        Train Loss: 0.283 | Train Acc: 89.98%
        Validation Loss: 0.260 | Validation Acc: 90.75%
Current ecoch:  4
        Train Loss: 0.259 | Train Acc: 90.70%
        Validation Loss: 0.229 | Validation Acc: 91.95%
Current ecoch:  5
        Train Loss: 0.238 | Train Acc: 91.49%
        Validation Loss: 0.255 | Validation Acc: 90.64%
Current ecoch:  6
        Train Loss: 0.225 | Train Acc: 92.05%
        Validation Loss: 0.212 | Validation Acc: 92.53%
Current ecoch:  7
        Train Loss: 0.209 | Train Acc: 92.53%
        Validation Loss: 0.419 | Validation Acc: 86.61%
Current ecoch:  8
        Train Loss: 0.196 | Train Acc: 93.07%
        Validation Loss: 0.202 | Validation Acc: 92.67%
Current ecoch:  9
        Train Loss: 0.185 | Train Acc: 93.38%
        Validation Loss: 0.261 | Validation Acc: 90.92%
Batch Size: 64
Current ecoch:  0
        Train Loss: 0.631 | Train Acc: 77.26%
        Validation Loss: 0.414 | Validation Acc: 84.62%
Current ecoch:  1
        Train Loss: 0.385 | Train Acc: 86.22%
        Validation Loss: 0.345 | Validation Acc: 86.93%
Current ecoch:  2
        Train Loss: 0.321 | Train Acc: 88.59%
        Validation Loss: 0.286 | Validation Acc: 89.76%
Current ecoch:  3
        Train Loss: 0.281 | Train Acc: 89.96%
        Validation Loss: 0.275 | Validation Acc: 90.00%
Current ecoch:  4
        Train Loss: 0.255 | Train Acc: 91.05%
        Validation Loss: 0.247 | Validation Acc: 91.04%
Current ecoch:  5
        Train Loss: 0.240 | Train Acc: 91.55%
        Validation Loss: 0.243 | Validation Acc: 91.33%
Current ecoch:  6
        Train Loss: 0.224 | Train Acc: 92.03%
        Validation Loss: 0.236 | Validation Acc: 91.47%
Current ecoch:  7
        Train Loss: 0.208 | Train Acc: 92.65%
        Validation Loss: 0.201 | Validation Acc: 92.85%
Current ecoch:  8
        Train Loss: 0.197 | Train Acc: 92.94%
        Validation Loss: 0.201 | Validation Acc: 92.53%
Current ecoch:  9
        Train Loss: 0.188 | Train Acc: 93.26%
        Validation Loss: 0.214 | Validation Acc: 92.45%
Batch Size: 128
Current ecoch:  0
        Train Loss: 0.643 | Train Acc: 76.19%
        Validation Loss: 0.574 | Validation Acc: 79.48%
Current ecoch:  1
        Train Loss: 0.385 | Train Acc: 86.25%
        Validation Loss: 0.383 | Validation Acc: 85.82%
Current ecoch:  2
        Train Loss: 0.318 | Train Acc: 88.60%
        Validation Loss: 0.314 | Validation Acc: 88.75%
Current ecoch:  3
        Train Loss: 0.287 | Train Acc: 89.81%
        Validation Loss: 0.260 | Validation Acc: 90.48%
Current ecoch:  4
        Train Loss: 0.254 | Train Acc: 90.99%
        Validation Loss: 0.248 | Validation Acc: 90.61%
Current ecoch:  5
        Train Loss: 0.235 | Train Acc: 91.75%
        Validation Loss: 0.218 | Validation Acc: 91.70%
Current ecoch:  6
        Train Loss: 0.220 | Train Acc: 92.19%
        Validation Loss: 0.224 | Validation Acc: 92.03%
Current ecoch:  7
        Train Loss: 0.208 | Train Acc: 92.58%
        Validation Loss: 0.224 | Validation Acc: 92.08%
Current ecoch:  8
        Train Loss: 0.196 | Train Acc: 92.95%
        Validation Loss: 0.219 | Validation Acc: 92.51%
Current ecoch:  9
        Train Loss: 0.185 | Train Acc: 93.46%
        Validation Loss: 0.198 | Validation Acc: 93.00%
Batch Size: 256
Current ecoch:  0
        Train Loss: 0.628 | Train Acc: 77.12%
        Validation Loss: 0.399 | Validation Acc: 84.94%
Current ecoch:  1
        Train Loss: 0.375 | Train Acc: 86.57%
        Validation Loss: 0.306 | Validation Acc: 88.48%
Current ecoch:  2
        Train Loss: 0.317 | Train Acc: 88.63%
        Validation Loss: 0.282 | Validation Acc: 90.28%
Current ecoch:  3
        Train Loss: 0.282 | Train Acc: 89.87%
        Validation Loss: 0.270 | Validation Acc: 90.19%
Current ecoch:  4
        Train Loss: 0.256 | Train Acc: 90.93%
        Validation Loss: 0.242 | Validation Acc: 91.37%
Current ecoch:  5
        Train Loss: 0.239 | Train Acc: 91.48%
        Validation Loss: 0.222 | Validation Acc: 92.02%
Current ecoch:  6
        Train Loss: 0.221 | Train Acc: 92.12%
        Validation Loss: 0.242 | Validation Acc: 90.77%
```

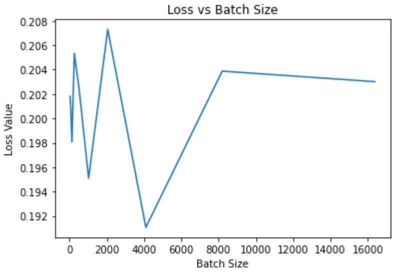
```
Current ecoch:  7
        Train Loss: 0.208 | Train Acc: 92.57%
        Validation Loss: 0.215 | Validation Acc: 92.21%
Current ecoch:  8
        Train Loss: 0.196 | Train Acc: 93.04%
        Validation Loss: 0.205 | Validation Acc: 92.68%
Current ecoch:  9
        Train Loss: 0.182 | Train Acc: 93.49%
        Validation Loss: 0.220 | Validation Acc: 92.36%
Batch Size: 512
Current ecoch:  0
        Train Loss: 0.632 | Train Acc: 77.02%
        Validation Loss: 0.421 | Validation Acc: 83.26%
Current ecoch:  1
        Train Loss: 0.386 | Train Acc: 86.16%
        Validation Loss: 0.318 | Validation Acc: 88.53%
Current ecoch:  2
        Train Loss: 0.327 | Train Acc: 88.31%
        Validation Loss: 0.271 | Validation Acc: 89.96%
Current ecoch:  3
        Train Loss: 0.288 | Train Acc: 89.61%
        Validation Loss: 0.311 | Validation Acc: 89.19%
Current ecoch:  4
        Train Loss: 0.264 | Train Acc: 90.53%
        Validation Loss: 0.277 | Validation Acc: 89.63%
Current ecoch:  5
        Train Loss: 0.242 | Train Acc: 91.31%
        Validation Loss: 0.237 | Validation Acc: 91.72%
Current ecoch:  6
        Train Loss: 0.224 | Train Acc: 92.03%
        Validation Loss: 0.211 | Validation Acc: 92.51%
Current ecoch:  7
        Train Loss: 0.208 | Train Acc: 92.57%
        Validation Loss: 0.214 | Validation Acc: 92.26%
Current ecoch:  8
        Train Loss: 0.199 | Train Acc: 92.88%
        Validation Loss: 0.202 | Validation Acc: 92.73%
Current ecoch:  9
        Train Loss: 0.190 | Train Acc: 93.30%
        Validation Loss: 0.214 | Validation Acc: 92.20%
Batch Size: 1024
Current ecoch:  0
        Train Loss: 0.646 | Train Acc: 76.43%
        Validation Loss: 0.435 | Validation Acc: 83.52%
Current ecoch:  1
        Train Loss: 0.386 | Train Acc: 86.10%
        Validation Loss: 0.361 | Validation Acc: 87.22%
Current ecoch:  2
        Train Loss: 0.320 | Train Acc: 88.52%
        Validation Loss: 0.280 | Validation Acc: 89.54%
Current ecoch:  3
        Train Loss: 0.284 | Train Acc: 89.77%
        Validation Loss: 0.289 | Validation Acc: 90.05%
Current ecoch:  4
        Train Loss: 0.259 | Train Acc: 90.71%
        Validation Loss: 0.278 | Validation Acc: 89.47%
Current ecoch:  5
        Train Loss: 0.240 | Train Acc: 91.51%
        Validation Loss: 0.223 | Validation Acc: 91.93%
Current ecoch:  6
        Train Loss: 0.223 | Train Acc: 92.05%
        Validation Loss: 0.231 | Validation Acc: 91.32%
Current ecoch:  7
        Train Loss: 0.211 | Train Acc: 92.53%
        Validation Loss: 0.210 | Validation Acc: 92.07%
Current ecoch:  8
        Train Loss: 0.196 | Train Acc: 92.97%
        Validation Loss: 0.195 | Validation Acc: 92.60%
Current ecoch:  9
        Train Loss: 0.186 | Train Acc: 93.47%
        Validation Loss: 0.196 | Validation Acc: 93.04%
Batch Size: 2048
Current ecoch:  0
        Train Loss: 0.635 | Train Acc: 76.71%
        Validation Loss: 0.404 | Validation Acc: 85.68%
Current ecoch:  1
        Train Loss: 0.384 | Train Acc: 86.31%
        Validation Loss: 0.305 | Validation Acc: 88.94%
Current ecoch:  2
        Train Loss: 0.325 | Train Acc: 88.45%
        Validation Loss: 0.316 | Validation Acc: 89.17%
Current ecoch:  3
        Train Loss: 0.288 | Train Acc: 89.38%
        Validation Loss: 0.242 | Validation Acc: 91.32%
Current ecoch:  4
        Train Loss: 0.258 | Train Acc: 90.74%
        Validation Loss: 0.233 | Validation Acc: 91.49%
Current ecoch:  5
        Train Loss: 0.241 | Train Acc: 91.42%
        Validation Loss: 0.252 | Validation Acc: 90.20%
Current ecoch:  6
        Train Loss: 0.223 | Train Acc: 92.03%
        Validation Loss: 0.212 | Validation Acc: 92.38%
Current ecoch:  7
        Train Loss: 0.209 | Train Acc: 92.48%
        Validation Loss: 0.221 | Validation Acc: 92.08%
Current ecoch:  8
        Train Loss: 0.199 | Train Acc: 92.85%
        Validation Loss: 0.207 | Validation Acc: 92.54%
Current ecoch:  9
        Train Loss: 0.186 | Train Acc: 93.32%
        Validation Loss: 0.210 | Validation Acc: 92.48%
Batch Size: 4096
Current ecoch:  0
        Train Loss: 0.659 | Train Acc: 75.85%
        Validation Loss: 0.433 | Validation Acc: 83.73%
Current ecoch:  1
        Train Loss: 0.389 | Train Acc: 85.99%
        Validation Loss: 0.383 | Validation Acc: 85.73%
Current ecoch:  2
        Train Loss: 0.320 | Train Acc: 88.43%
        Validation Loss: 0.287 | Validation Acc: 89.52%
Current ecoch:  3
        Train Loss: 0.280 | Train Acc: 89.92%
        Validation Loss: 0.295 | Validation Acc: 89.50%
```

```
Current ecoch:  4
        Train Loss: 0.256 | Train Acc: 90.93%
        Validation Loss: 0.233 | Validation Acc: 91.78%
Current ecoch:  5
        Train Loss: 0.234 | Train Acc: 91.62%
        Validation Loss: 0.250 | Validation Acc: 91.10%
Current ecoch:  6
        Train Loss: 0.221 | Train Acc: 92.16%
        Validation Loss: 0.224 | Validation Acc: 92.10%
Current ecoch:  7
        Train Loss: 0.206 | Train Acc: 92.73%
        Validation Loss: 0.204 | Validation Acc: 92.49%
Current ecoch:  8
        Train Loss: 0.197 | Train Acc: 92.99%
        Validation Loss: 0.212 | Validation Acc: 91.90%
Current ecoch:  9
        Train Loss: 0.186 | Train Acc: 93.38%
        Validation Loss: 0.191 | Validation Acc: 93.12%
Batch Size: 8192
Current ecoch:  0
        Train Loss: 0.626 | Train Acc: 77.26%
        Validation Loss: 0.391 | Validation Acc: 86.04%
Current ecoch:  1
        Train Loss: 0.381 | Train Acc: 86.21%
        Validation Loss: 0.370 | Validation Acc: 87.18%
Current ecoch:  2
        Train Loss: 0.317 | Train Acc: 88.74%
        Validation Loss: 0.539 | Validation Acc: 80.90%
Current ecoch:  3
        Train Loss: 0.282 | Train Acc: 89.98%
        Validation Loss: 0.257 | Validation Acc: 91.40%
Current ecoch:  4
        Train Loss: 0.254 | Train Acc: 91.00%
        Validation Loss: 0.240 | Validation Acc: 91.36%
Current ecoch:  5
        Train Loss: 0.235 | Train Acc: 91.59%
        Validation Loss: 0.233 | Validation Acc: 91.46%
Current ecoch:  6
        Train Loss: 0.221 | Train Acc: 92.24%
        Validation Loss: 0.214 | Validation Acc: 92.25%
Current ecoch:  7
        Train Loss: 0.206 | Train Acc: 92.61%
        Validation Loss: 0.212 | Validation Acc: 92.13%
Current ecoch:  8
        Train Loss: 0.196 | Train Acc: 93.06%
        Validation Loss: 0.204 | Validation Acc: 92.68%
Current ecoch:  9
        Train Loss: 0.184 | Train Acc: 93.34%
        Validation Loss: 0.210 | Validation Acc: 92.40%
Batch Size: 16384
Current ecoch:  0
        Train Loss: 0.647 | Train Acc: 76.25%
        Validation Loss: 0.436 | Validation Acc: 83.96%
Current ecoch:  1
        Train Loss: 0.395 | Train Acc: 85.73%
        Validation Loss: 0.388 | Validation Acc: 86.21%
Current ecoch:  2
        Train Loss: 0.324 | Train Acc: 88.40%
        Validation Loss: 0.294 | Validation Acc: 89.35%
Current ecoch:  3
        Train Loss: 0.286 | Train Acc: 89.76%
        Validation Loss: 0.285 | Validation Acc: 89.26%
Current ecoch:  4
        Train Loss: 0.257 | Train Acc: 90.76%
        Validation Loss: 0.249 | Validation Acc: 91.47%
Current ecoch:  5
        Train Loss: 0.240 | Train Acc: 91.49%
        Validation Loss: 0.225 | Validation Acc: 91.82%
Current ecoch:  6
        Train Loss: 0.222 | Train Acc: 92.18%
        Validation Loss: 0.238 | Validation Acc: 91.35%
Current ecoch:  7
        Train Loss: 0.210 | Train Acc: 92.60%
        Validation Loss: 0.212 | Validation Acc: 92.44%
Current ecoch:  8
        Train Loss: 0.198 | Train Acc: 92.93%
        Validation Loss: 0.263 | Validation Acc: 89.94%
Current ecoch:  9
        Train Loss: 0.184 | Train Acc: 93.40%
        Validation Loss: 0.203 | Validation Acc: 92.71%
```

In [ ]:

```python
def plot_losses_batch_version(losses):
    plt.plot(list(losses.keys()), list(losses.values()))
    plt.xlabel('Batch Size')
    plt.ylabel('Loss Value')
    plt.title('Loss vs Batch Size')
    plt.show()

plot_losses_batch_version(batch_size_to_loss)
```


Loss vs Batch Size
Loss Value
Batch Size