

Fashion MNIST

Training MLP networks with PyTorch
on Fashion MNIST dataset

Gradients w.r.t Inputs

Q: Calculate the standard deviation of the gradients for each trail and sort the highest 10 std of the input.

```
for epoch in range(num_epochs):  
  
    grad_collect_batch=torch.zeros(batch_size, input_size).cuda()  
  
    for i, data in enumerate(train_loader,0):  
        images, labels = data  
        images= images.view(-1, 1 * 28 * 28).cuda() # read data by certain format  
        images, labels = Variable(images, requires_grad=True), Variable(labels.cuda())  
        optimizer.zero_grad()  
        outputs = net(images)  
        loss = criterion(outputs, labels)  
        loss.backward()  
        optimizer.step()  
  
        # append gradient  
        gradient=images.grad  
        grad_collect_batch+=gradient
```

Gradients w.r.t Inputs

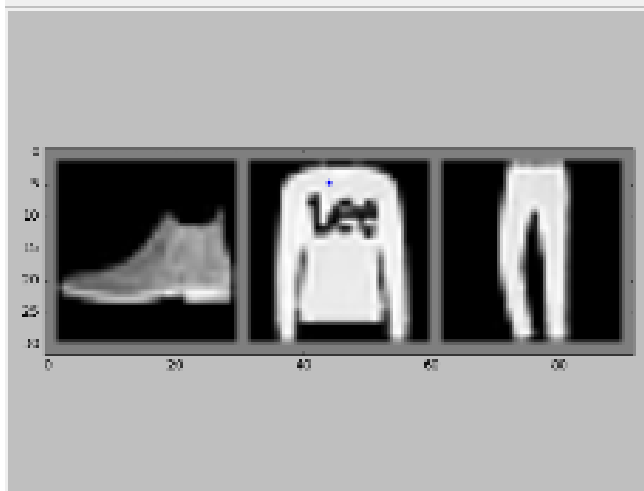
A:

Calculating the standard deviation of the gradients with respect to inputs can be used to find the potential key inputs. Each training iteration tends to minimize the loss function. When the loss function is partial differential to the input variables, at each iteration, the gradient of variable represents the change of each input variable to minimize the loss function.

The sequence of inputs with the highest 10 standard deviation of the gradients for each trail are [454,397,38,37,453 ,17 ,45,11,425 ,46]. Those numbers stand for the location on the picture resulting in drastic change as each trail increasing minimize the loss function. In other words, this location is the key potential feature when the NNP model is created.

Fashion MNIST Datasets

Fashion MNIST is a dataset of Zalando's article images—consisting of a training set of 60,000 examples and a test set of 10,000 examples. Each example is a 28x28 grayscale image, associated with a label from 10 classes.



Prediction is Ankle boot

Prediction is Pullover

Prediction is Trouser