Author: Kunjan Mhaske

----------FASHION-MNIST MODEL----------

Dataset: Fashion-MNIST

Fashion-MNIST is a dataset of Zalando's article images which consists

- Training set of 60,000 examples
- Test set of 10,000 examples
- Each example is a 28 x 28 grayscale image: 784 pixels valued between 0 and 255
- Each associated with a label from 10 classes
- Each row is a separate image
- Column 1 is the class label.
- Remaining columns are pixel numbers (784 total).
- Each value is the darkness of the pixel (1 to 255)

Classes:

0 - T-shirt/top, 1 – Trouser, 2 – Pullover, 3 – Dress, 4 – Coat, 5 – Sandal, 6 – Shirt, 7 -Sneaker, 8 – Bag, 9 - Ankle boot.

Model: Neural Network containing 2D-Convolution and Linear Transformation

2D-Convolution: Applies a 2D convolution over an input signal composed of several input planes. In the simplest case, the output value of the layer with input size (N, C_{in} , H, W) and output (N, C_{out} , H_{out} , W_{out}) can be precisely described as:

$$\text{out}(N_i, C_{\text{out}_j}) = \text{bias}(C_{\text{out}_j}) + \sum_{k=0}^{C_{\text{in}}-1} \text{weight}(C_{\text{out}_j}, k) \star \text{input}(N_i, k)$$

where $\star$ is the valid 2D cross-correlation operator, N is a batch size, C denotes a number of channels, HH is a height of input planes in pixels, and W is width in pixels.

```
self.conv1 = nn.Conv2d(1,6,5)
self.conv2 = nn.Conv2d(6,16,5)
```

Linear Transformation:

Linear Transformation applies a linear transformation to the incoming data: $y = x.A^T + b$

```
self.fc1 = nn.Linear(16*5*5, 120)
self.fc2 = nn.Linear(120, 84)
self.fc3 = nn.Linear(84, 10)
```

Loss and Activation Functions

Loss Function:

Cross Entropy Loss : Cross entropy quantifies the difference between two probability distribution. Our model predicts a model distribution of {p1, p2, p3 ….., p10} (where p1+p2+p3+…..+p10 = 1). We use cross-entropy to compare this with the true distribution {y1, y2, y3, …, y10}

The losses are averaged across observations for each minibatch.

```
criterion = nn.CrossEntropyLoss()
```

$$\text{loss}(x, class) = weight[class]\left(-x[class] + \log\left(\sum_j \exp(x[j])\right)\right)$$

Activation Functions:

ReLu - Rectified linear unit. Applies the rectified linear unit function element-wise.

ReLU(x)= max(0, x)

```
x = F.avg_pool2d(F.relu(self.conv1(x)), (2,2))
x = F.avg_pool2d(F.relu(self.conv2(x)), 2)
#
x = F.relu(self.fc1(x))
x = F.relu(self.fc2(x))
```

Softmax - Applies the Softmax function to an n-dimensional input Tensor rescaling them so that the elements of the n-dimensional output Tensor lie in the range (0,1) and sum to 1.

$$\text{Softmax}(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)}$$

```
return F.softmax(x,dim=1)
```

Optimizer: Adam optimizer - *adaptive moment estimation*

Adam is an optimization algorithm that can used instead of the classical stochastic gradient descent procedure to update network weights iterative based in training data.

- Adam is a replacement optimization algorithm for stochastic gradient descent for training deep learning models.
- Adam combines the best properties of the AdaGrad and RMSProp algorithms to provide an optimization algorithm that can handle sparse gradients on noisy problems.
- Adam is relatively easy to configure where the default configuration parameters do well on most problems.

```
optimizer = optim.Adam(net.parameters(), lr=0.001)
```

RESULTS

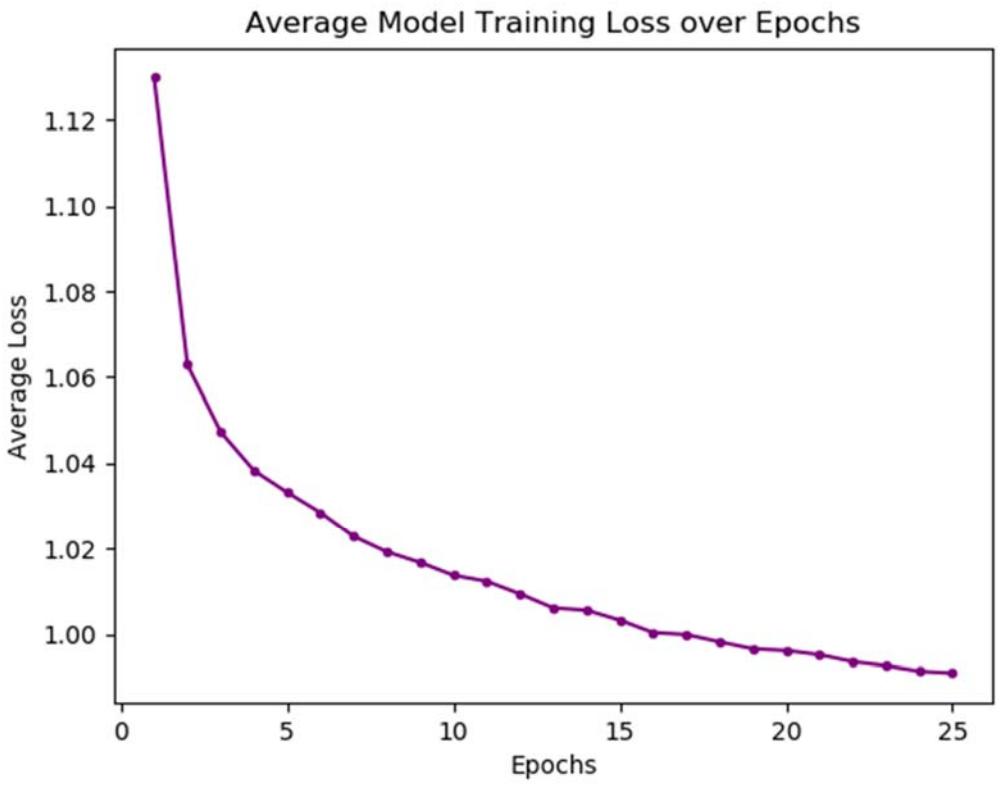
Experimental Prediction Comparisons: EPOCHS 25, LR 0.001**,** Saved in file **Output.txt**

Batch Size	Overall Accuracy	Class-wise Accuracy
32	88.72 %	T-shirt/top : 83 % Trouser : 96 % Pullover : 84 % Dress : 90 % Coat : 82 % Sandal : 96 % Shirt : 60 % Sneaker : 96 % Bag : 98 % Ankle boot : 93 %
64	88.94 %	T-shirt/top : 86 % Trouser : 94 % Pullover : 88 % Dress : 90 % Coat : 81 % Sandal : 98 % Shirt : 50 % Sneaker : 98 % Bag : 94 % Ankle boot : 91 %
100	**89.76 %**	**T-shirt/top : 91 %** **Trouser : 100 %** **Pullover : 85 %** **Dress : 92 %** **Coat : 87 %** **Sandal : 91 %** **Shirt : 75 %** **Sneaker : 96 %** **Bag : 94 %** **Ankle boot : 97 %**
256	87.76 %	T-shirt/top : 80 % Trouser : 100 % Pullover : 76 % Dress : 85 % Coat : 88 % Sandal : 100 % Shirt : 57 % Sneaker : 95 % Bag : 93 % Ankle boot : 100 %

Best Execution Result: Batch Size = 100, LR = 0.001, Epochs 25

```
Device cuda 0
Properties: _CudaDeviceProperties(name='GeForce 840M', major=5, minor=0,
total_memory=4096MB, multi_processor_count=3)
Optimizer: Adam
Training Start.....
Batch size: 100
Epoch: 1/25, Approximated Loss: 1.129848426663125
Epoch: 2/25, Approximated Loss: 1.0628678306316999
Epoch: 3/25, Approximated Loss: 1.0472933776605091
Epoch: 4/25, Approximated Loss: 1.0382859022411237
Epoch: 5/25, Approximated Loss: 1.0331926876351063
Epoch: 6/25, Approximated Loss: 1.0284631376968758
Epoch: 7/25, Approximated Loss: 1.0228337952777695
Epoch: 8/25, Approximated Loss: 1.0192152504732668
Epoch: 9/25, Approximated Loss: 1.016727340132348
Epoch: 10/25, Approximated Loss: 1.0137242532972337
Epoch: 11/25, Approximated Loss: 1.012320290380252
Epoch: 12/25, Approximated Loss: 1.0093821302422084
Epoch: 13/25, Approximated Loss: 1.0061071088245355
Epoch: 14/25, Approximated Loss: 1.0055807673689396
Epoch: 15/25, Approximated Loss: 1.0032560717207135
Epoch: 16/25, Approximated Loss: 1.0003648341147342
Epoch: 17/25, Approximated Loss: 0.9998924694040924
Epoch: 18/25, Approximated Loss: 0.9982057388716853
Epoch: 19/25, Approximated Loss: 0.9966053588667699
Epoch: 20/25, Approximated Loss: 0.9962048587829606
Epoch: 21/25, Approximated Loss: 0.9952492450573656
Epoch: 22/25, Approximated Loss: 0.9936388998174107
Epoch: 23/25, Approximated Loss: 0.9926942509291903
Epoch: 24/25, Approximated Loss: 0.9914221831039787
Epoch: 25/25, Approximated Loss: 0.9909889172591675
_____________________________Finished Training_____________________________
_____________________________________________________________________________
The sample comparisons between ground truth vs the predicted labels

Ground Truth:  T-shirt/top | Dress | Shirt | Sandal | Pullover | Pullover | Dress |
Trouser | T-shirt/top | Sandal
Predicted:  T-shirt/top | Dress | Shirt | Sandal | Pullover | Pullover | Dress |
Trouser | T-shirt/top | Sandal

Accuracy of network: 89.76 %

_____________________Accuracies per class_____________________________
Accuracy of T-shirt/top : 91 %
Accuracy of Trouser : 100 %
Accuracy of Pullover : 85 %
Accuracy of Dress : 92 %
Accuracy of Coat : 87 %
Accuracy of Sandal : 91 %
Accuracy of Shirt : 75 %
Accuracy of Sneaker : 96 %
Accuracy of Bag : 94 %
Accuracy of Ankle boot : 97 %
---------------Plotting the predicted results of some random images-----------
Figures saved in current working folder.
Done..

Process finished with exit code 0
```

Best Training Loss Values: Saved in file **Fmnist_training_loss_a1.png**



Best Sample Predictions: Saved in files **0.png** and **1.png**