

Data download and preprocessing

Prepare a custom dataset using PyTorch for FashionMNIST, removing even classes from the dataset, and creating a new dataset with only odd classes. The initial code does the following:

1. Defines a transform that normalizes the data.
2. Downloads the FashionMNIST dataset and applies the defined transform.
3. Removes even classes from the dataset using a list of odd classes.
4. Creates new subsets of the dataset that only contain odd classes.
5. Converts the subsets into a custom dataset using `torch.utils.data.TensorDataset`.
6. Creates data loaders for the train and test sets with the custom dataset.

Overall, the code is preparing a new dataset that is a subset of the original FashionMNIST dataset with only odd classes. The code is also converting the data into tensors and creating data loaders for easy use in PyTorch models.

Model 1

The code contains a pre-trained ResNet-18 model to accept grayscale input and changing the number of output classes to 5. The input is expected to have one channel (grayscale). The code also modifies the first convolutional layer of the model to accept one channel input by changing the input channels from 3 to 1. The commented out line shows how to modify the last fully connected layer (fc) of the model, which by default has 512 input features (output features from the last convolutional block) and 1000 output features (one for each of the original ImageNet classes). However, in this case, the number of output features is changed to 10, which would be appropriate if the original ImageNet model was being adapted for use with the 10-class MNIST dataset, for example. Since the output here is 5 classes, it is not necessary to modify the number of output features.

Model 2

This code defines a new class `ResNet18_model2` that inherits from `nn.Module` and contains a ResNet-18 model with modifications. Specifically, the first convolutional layer is modified to accept grayscale input, and the number of output classes in the final fully connected layer is set to `num_classes`, which is initialized to 5.

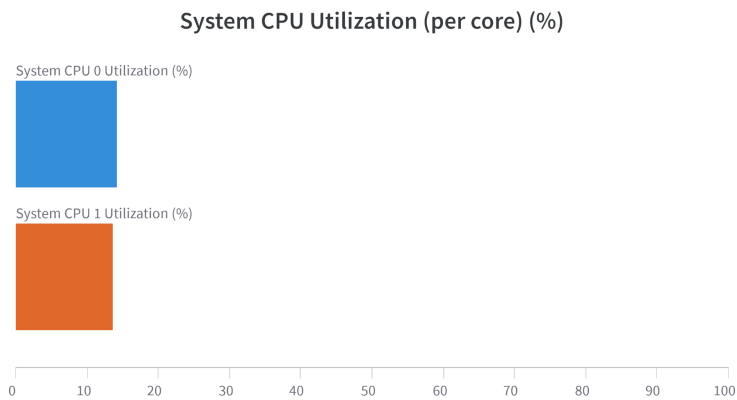
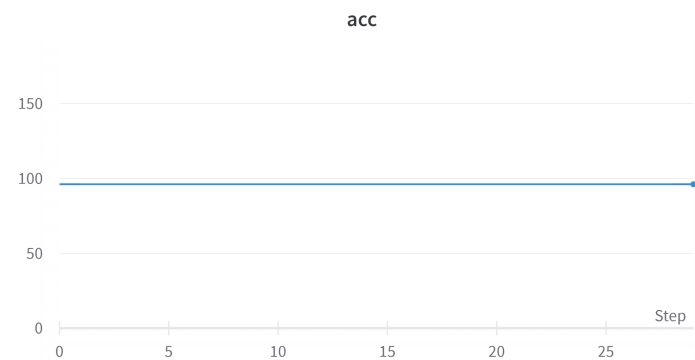
The forward method takes an input tensor `x` and passes it through the modified ResNet-18 model with a hyperbolic tangent activation function applied to the output. The output tensor is returned.

The code then creates an instance of the `ResNet18_model2` class and sends it to the GPU if available. It also defines a cross entropy loss criterion and a stochastic gradient descent (SGD) optimizer with a learning rate of 0.01 and momentum of 0.9.

Observations

Model 1:
Activation function - ReLU
Optimizer - Adam
Epochs - 30

Batch [400/469], Loss: 0.2475, Accuracy: 96.27%



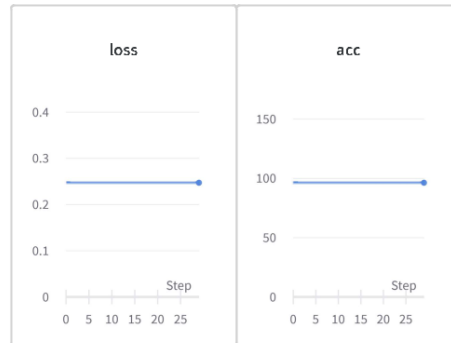
Report generated

Adam, ReLU: model 1

Add a description...

Princy Gautam

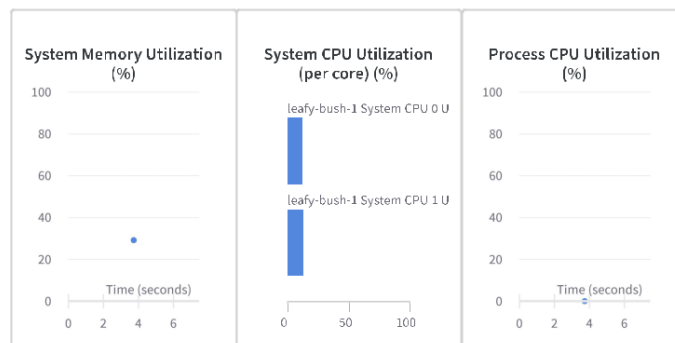
▼ Section 1



Import panel

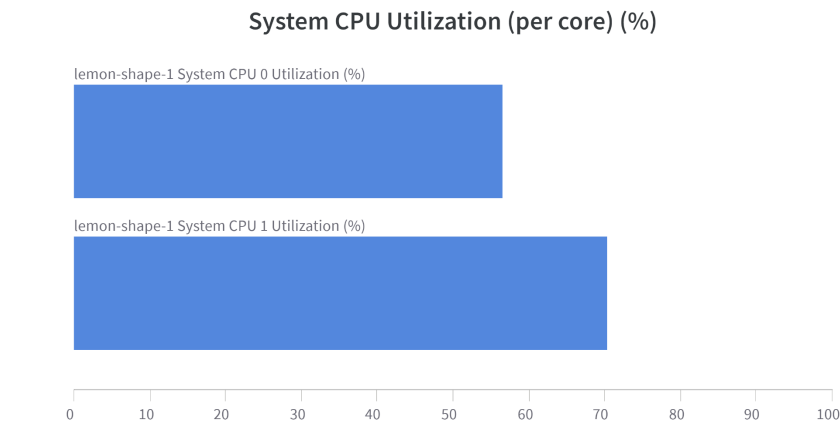
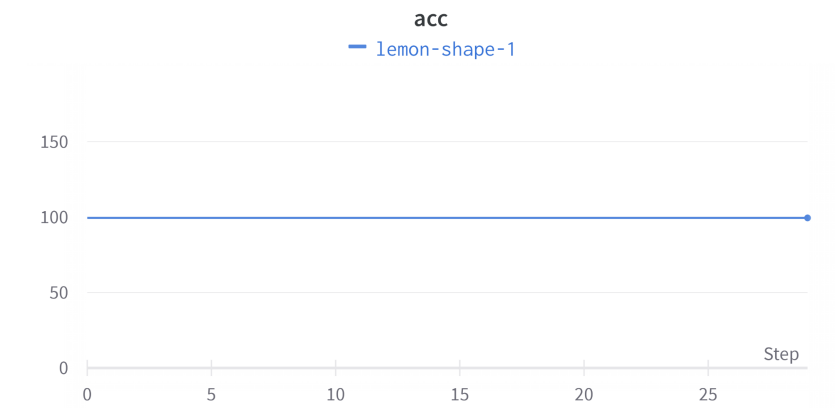
Add panel

<input checked="" type="checkbox"/>	Run set 1	:	
<div> ssup_a / FashionMNIST Project Filter Group </div>			
<input type="checkbox"/> Name (1 visualized)	State	Notes	User
<input checked="" type="checkbox"/> leafy-bush-1	finished	Add notes	princyg
<div>1-1 of 1 < ></div>			



Model 2:
Activation function - tanh
Optimizer - SGD
Epochs - 30

Batch [400/469], Loss: 0.4412, Accuracy: 99.52%



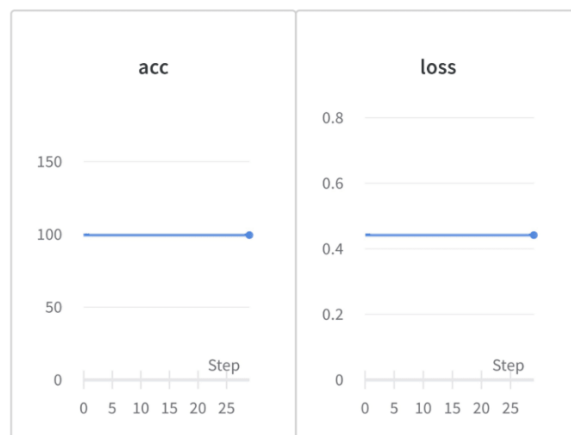
Report generated

SGD,tanh: Model 2

Add a description...

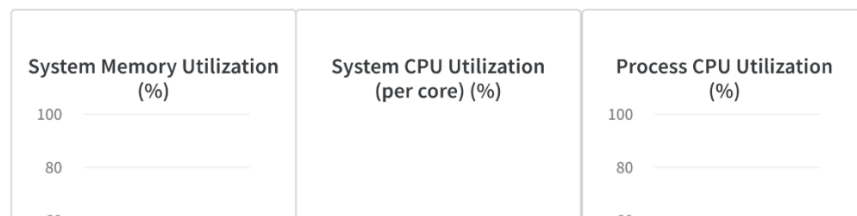
Princy Gautam

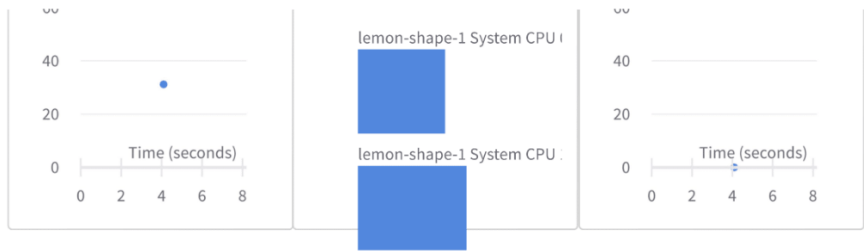
▼ Section 1



Import panel

Add panel





Import panel 50 Add panel 100

