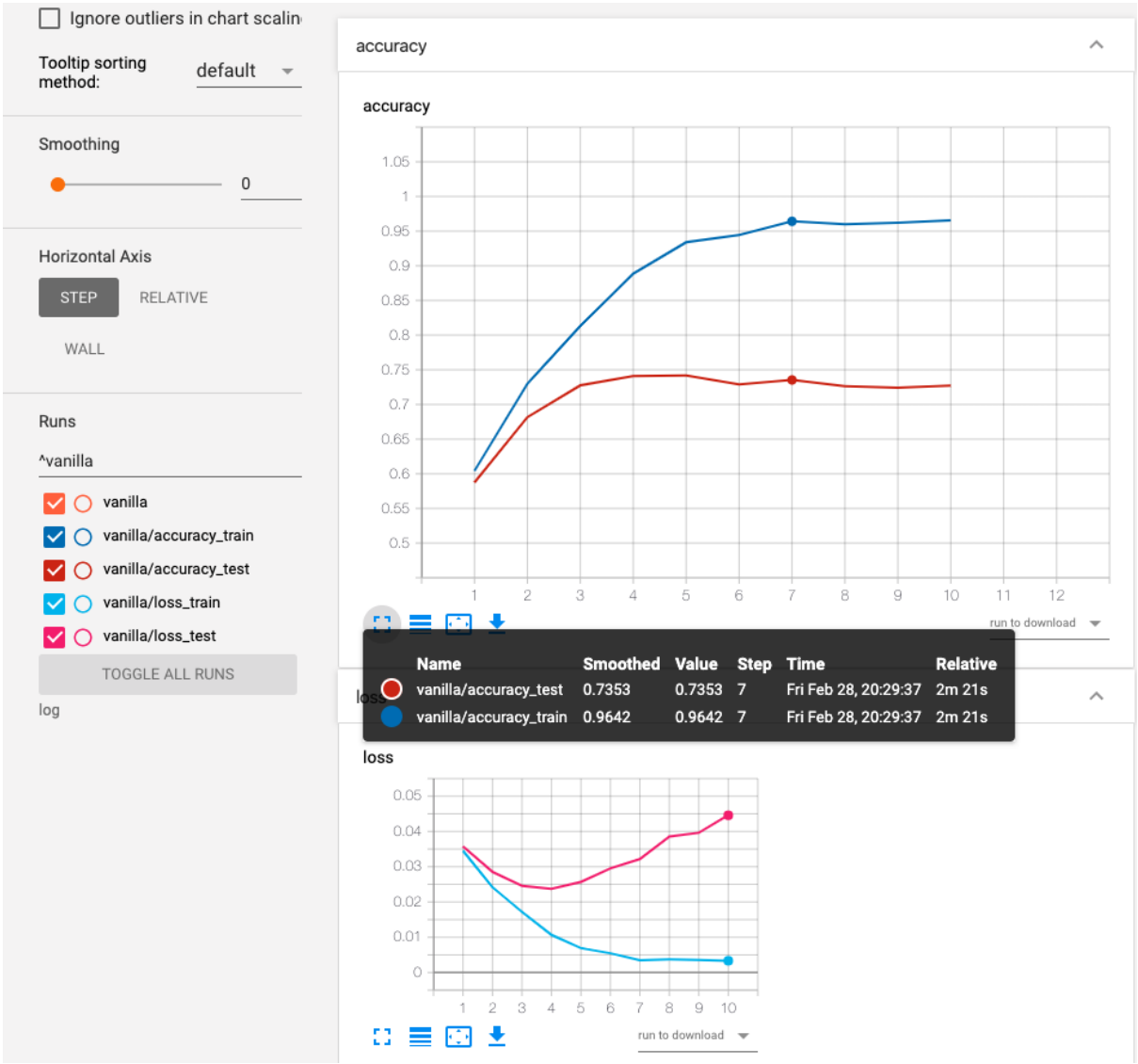
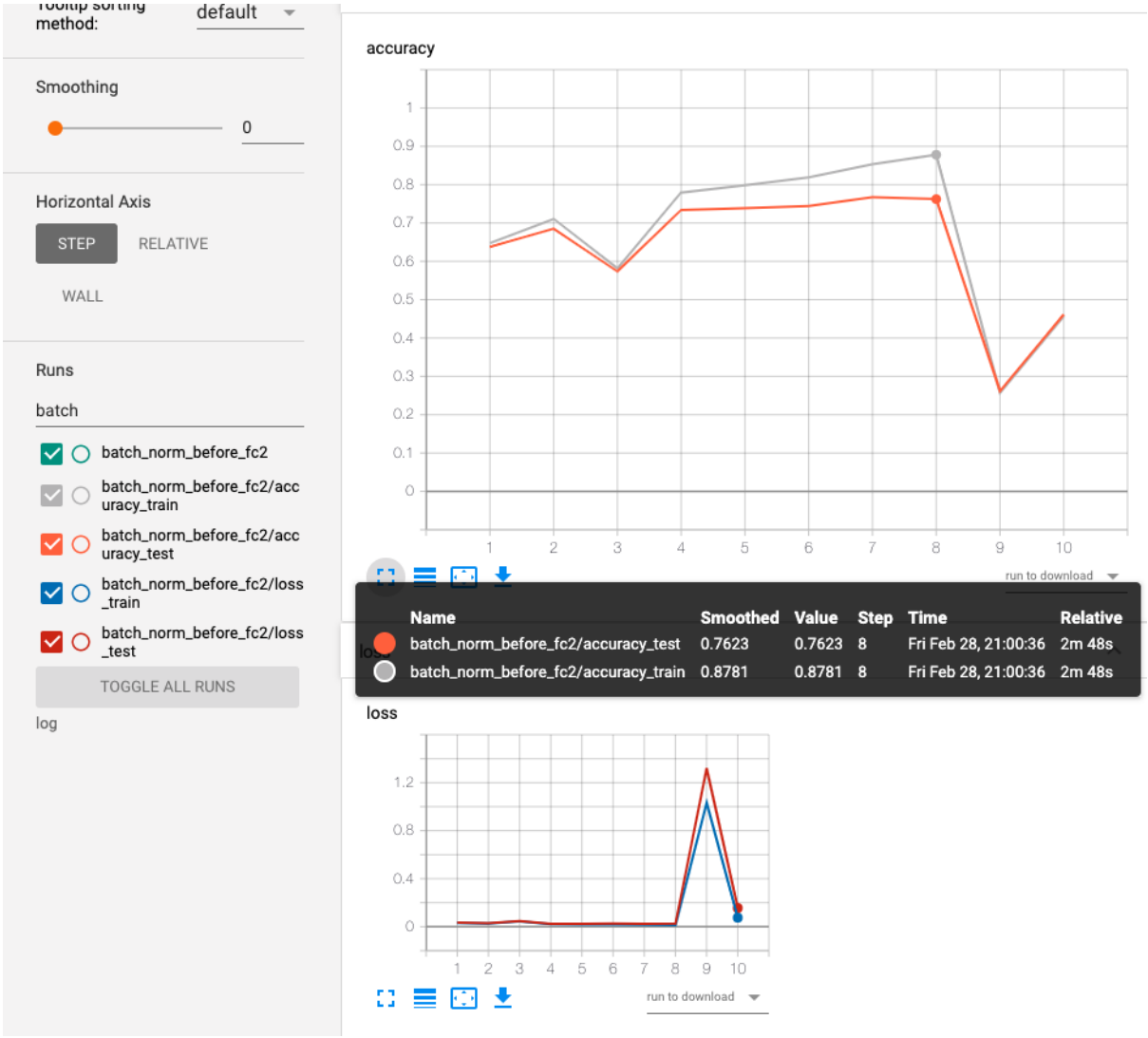
Rahul Damineni

CS535 – Assignment-3

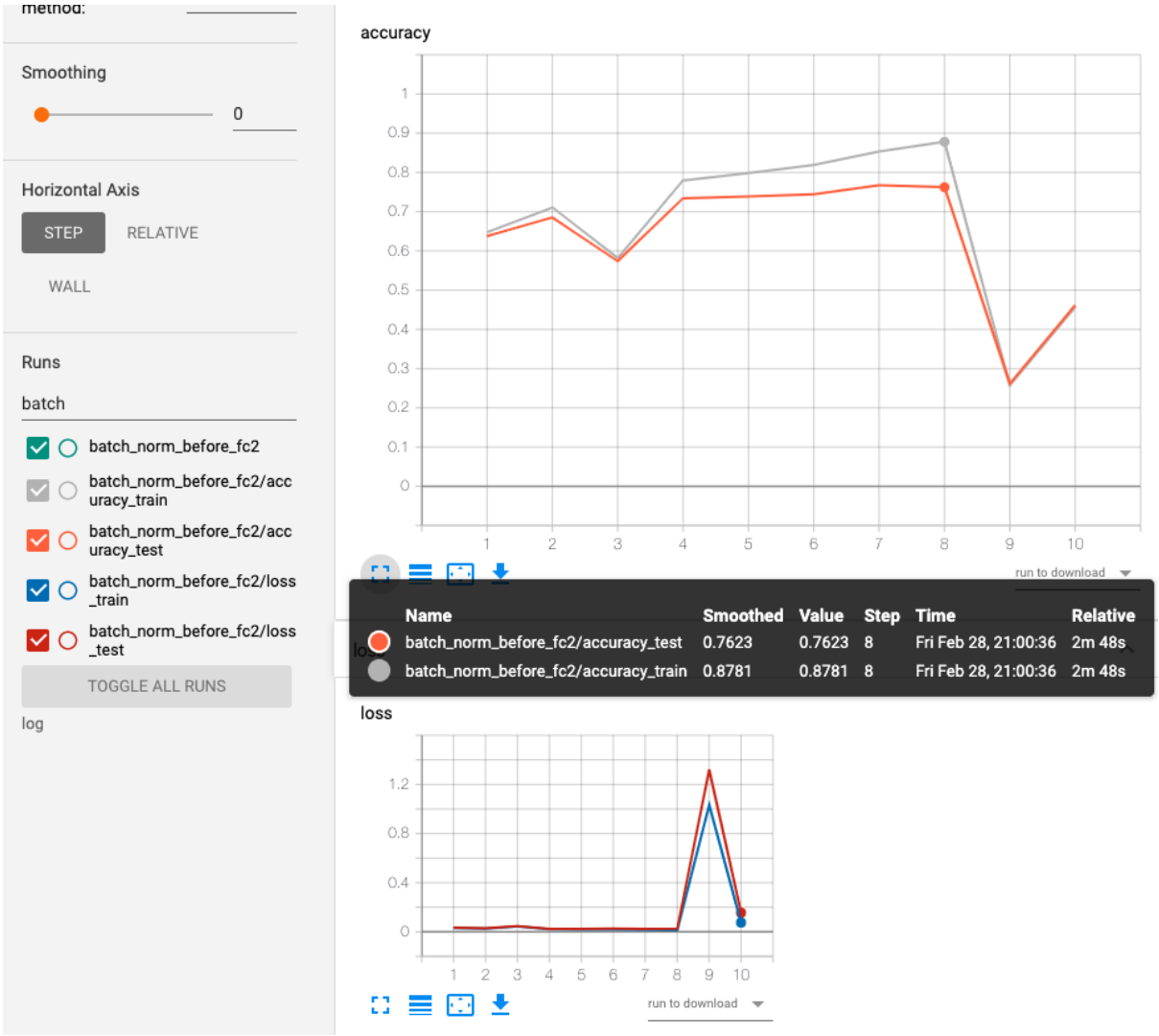
0. Vanilla (Default run)



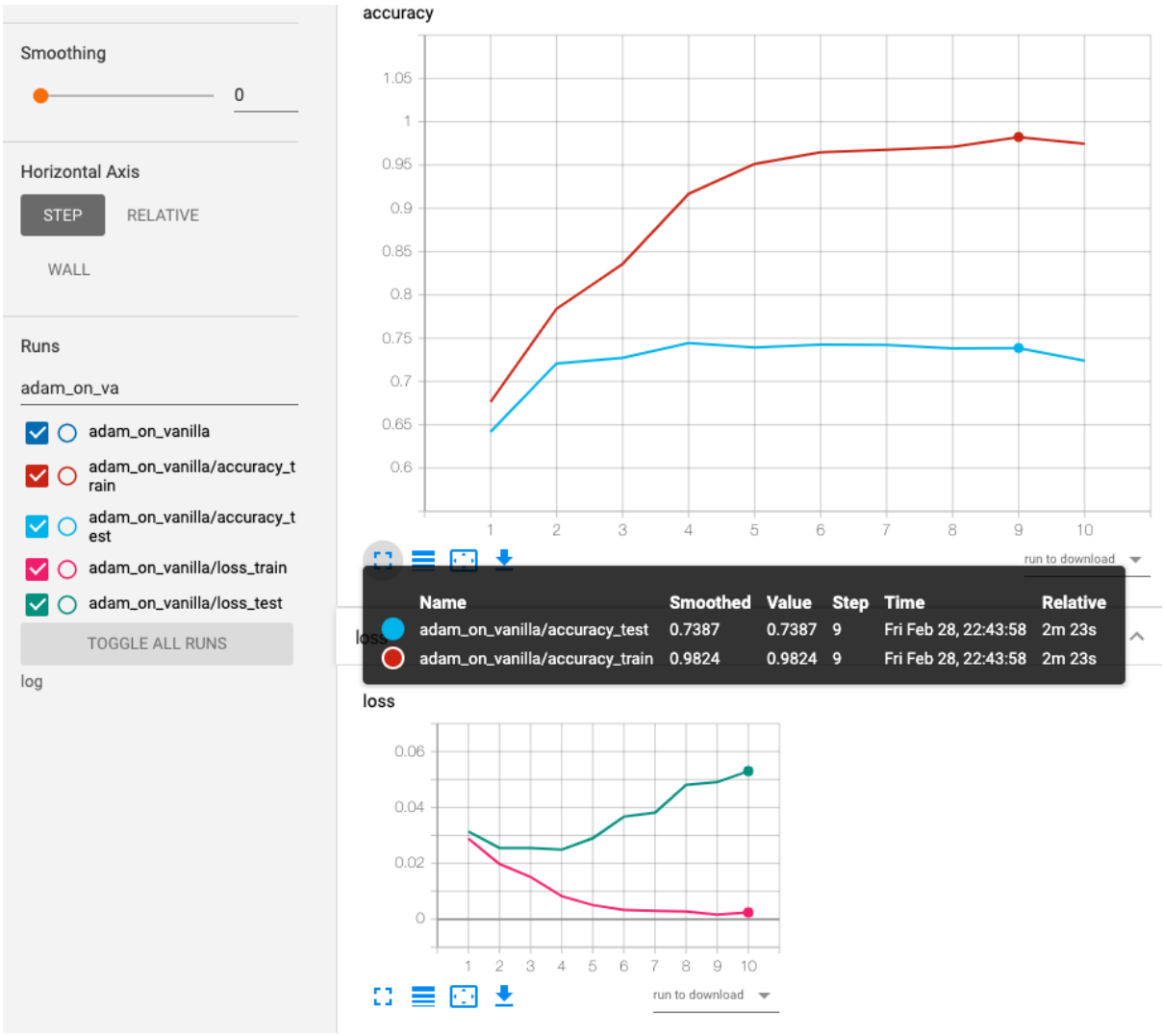
1. Batch Normalization before FC1



2. FC layer with 512 nodes before FC2

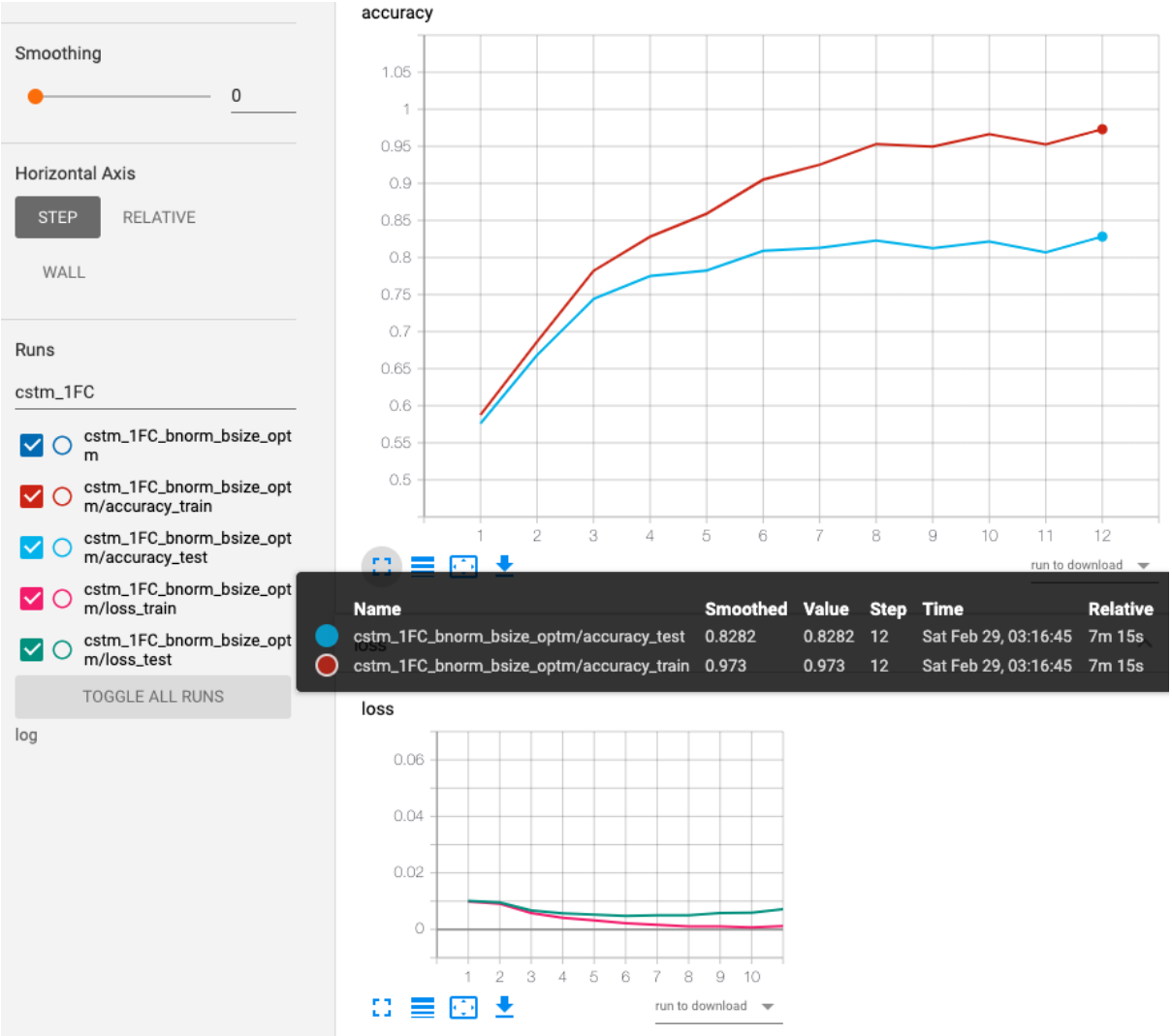


3. Adam optimizer for backprop

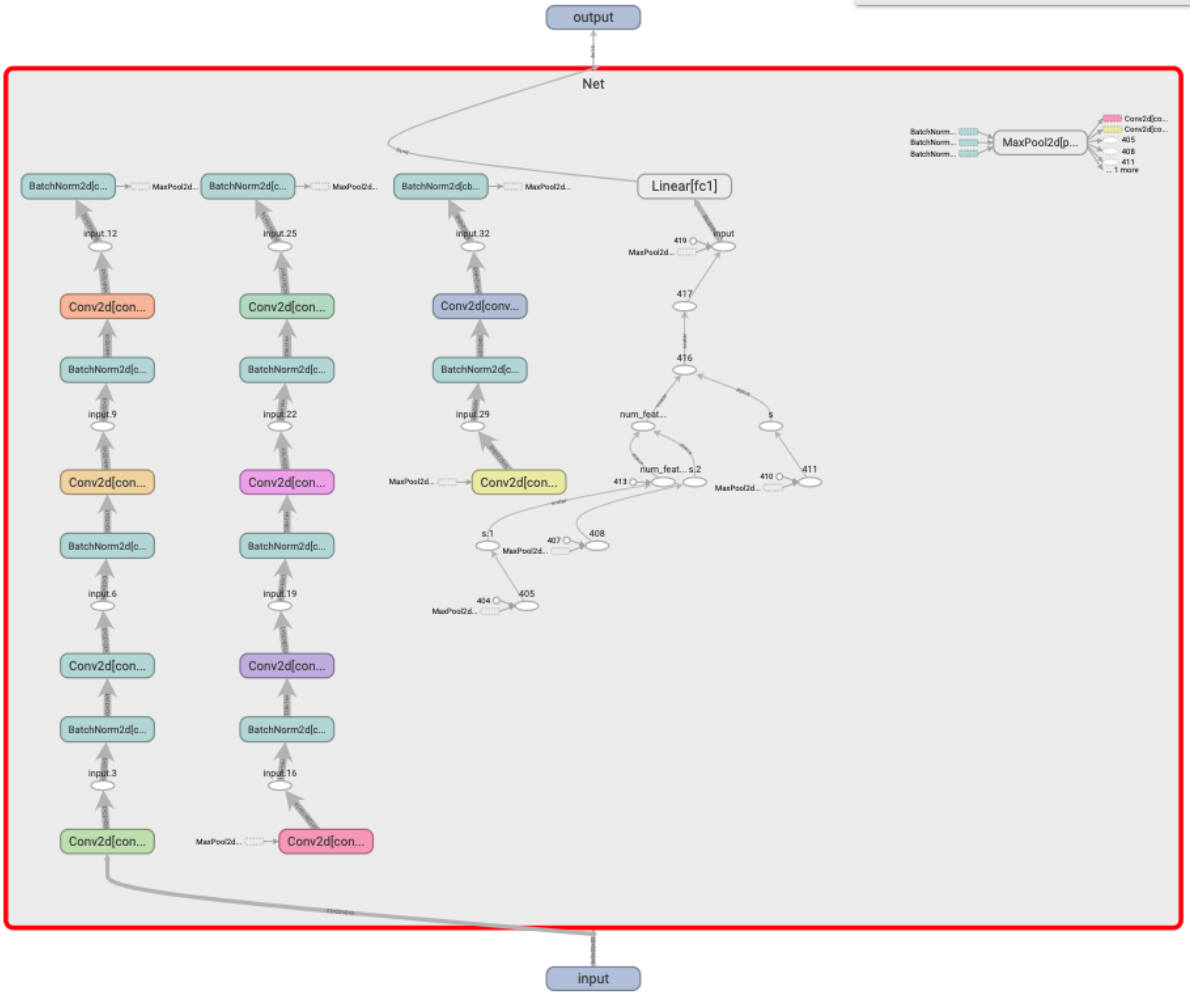


4. **After many experiments,**
    a. Removed FC2 and made FC1's output dimension 10
    b. Added 6 more conv layers making a total of 10 layers
    c. Applied Batch Norm layers each activated Conv layer's output
    d. Added MaxPool layers as and when out-volume had to be cut down
    e. Upscaled batch size to 128
    f. Increased epoch count to 12
    g. Selected torch.optim.AdamW as optimizer (weight decay)

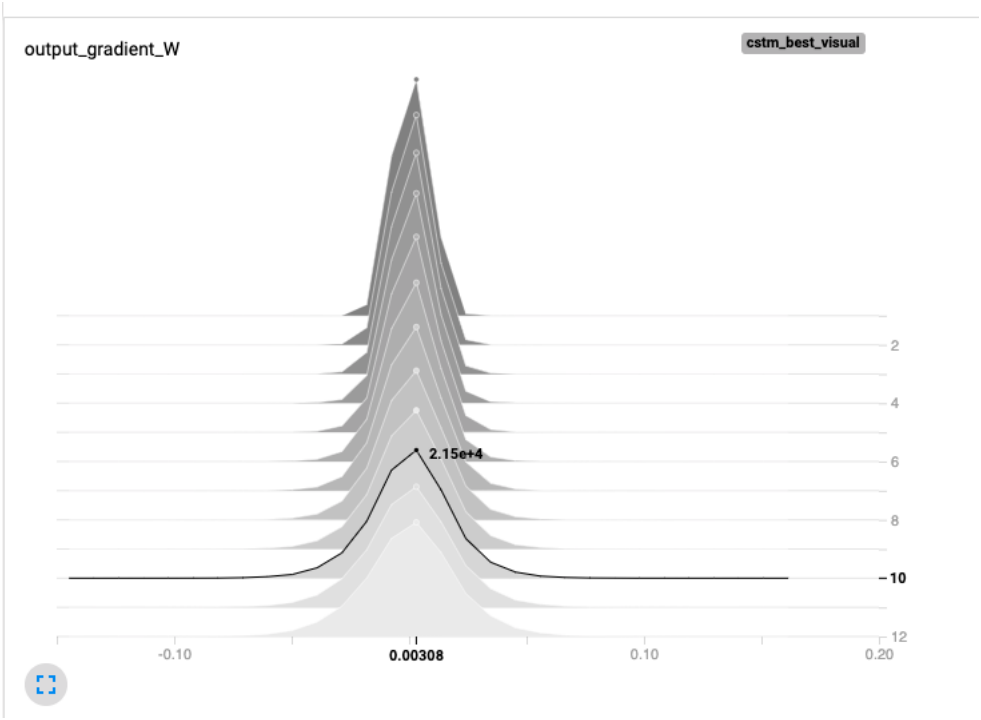
```
x = F.relu(self.conv1(x))
x = self.cbn1(x)
x = F.relu(self.conv2(x))
x = self.cbn2(x)
# x = self.pool(x)
#
x = F.relu(self.conv3(x))
x = self.cbn3(x)
x = F.relu(self.conv4(x))
x = self.cbn4(x)
x = self.pool(x)

x = F.relu(self.conv5(x))
x = self.cbn5(x)
x = F.relu(self.conv6(x))
x = self.cbn6(x)
# x = self.pool(x)

x = F.relu(self.conv7(x))
x = self.cbn7(x)
x = F.relu(self.conv8(x))
x = self.cbn8(x)
x = self.pool(x)

x = F.relu(self.conv9(x))
x = self.cbn9(x)
x = F.relu(self.conv10(x))
x = self.cbn10(x)
x = self.pool(x)

x = x.view(-1, self.num_flat_features(x))
x = self.fc1(x)
```

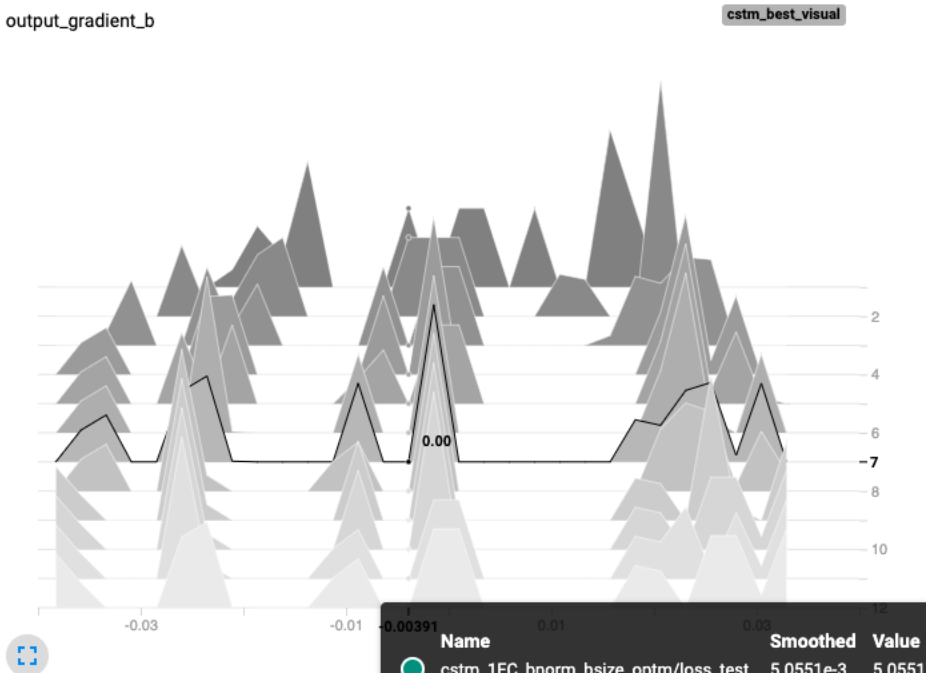

Smoothing
0
Horizontal Axis
STEP
RELATIVE
WALL
Runs
cstm_1FC
cstm_1FC_bnorm_bsize_optm
cstm_1FC_bnorm_bsize_optm/accuracy_train
cstm_1FC_bnorm_bsize_optm/accuracy_test
cstm_1FC_bnorm_bsize_optm/loss_train
cstm_1FC_bnorm_bsize_optm/loss_test
TOGGLE ALL RUNS
log
accuracy
1.05
1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0.55
0.5
1 2 3 4 5 6 7 8 9 10 11 12
run to download
Name Smoothed Value Step Time Relative
cstm_1FC_bnorm_bsize_optm/accuracy_test 0.8282 0.8282 12 Sat Feb 29, 03:16:45 7m 15s
cstm_1FC_bnorm_bsize_optm/accuracy_train 0.973 0.973 12 Sat Feb 29, 03:16:45 7m 15s
loss
0.06
0.04
0.02
0
1 2 3 4 5 6 7 8 9 10
run to download

5. **Visualization toolkit**
    a. **Model Graph**

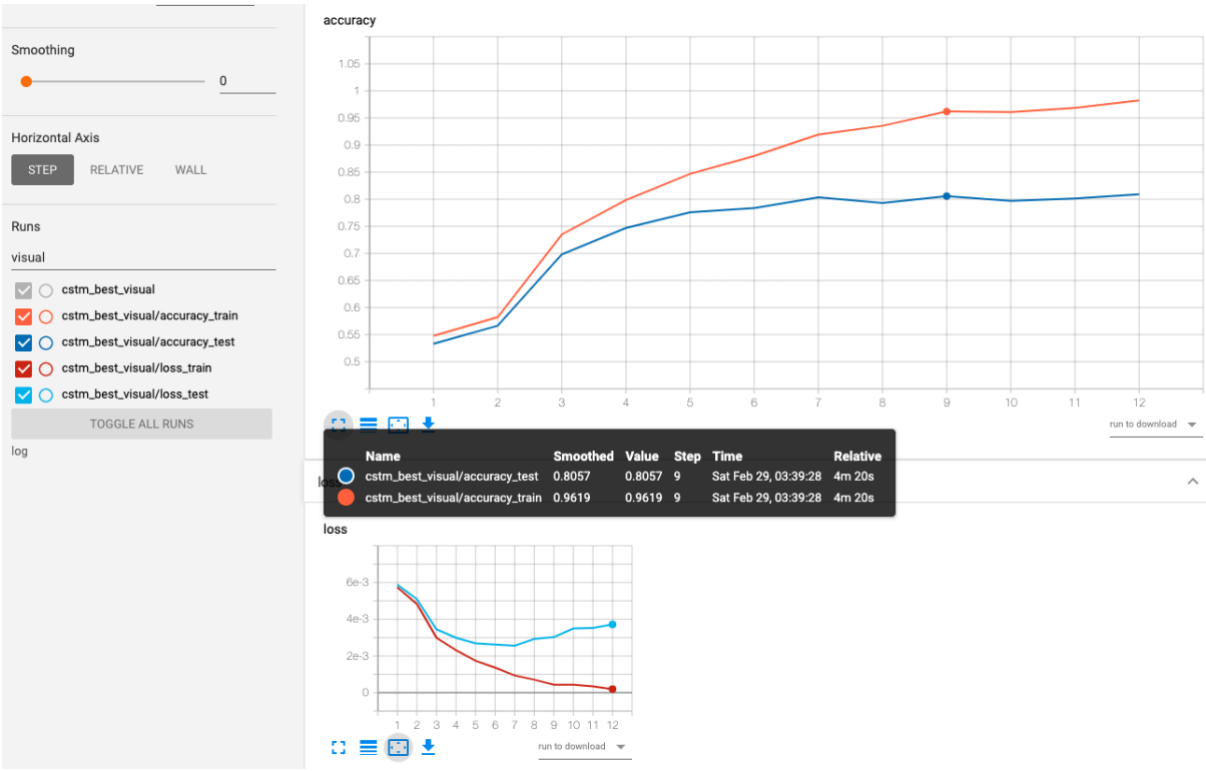


b. **Visualizing output gradients – See them going down sanely with epoch indicating proper training**





c. **And finally, monitoring train/test loss/accuracy per epoch**



I superposed multiple runs using the regex filter to compare b/w runs.