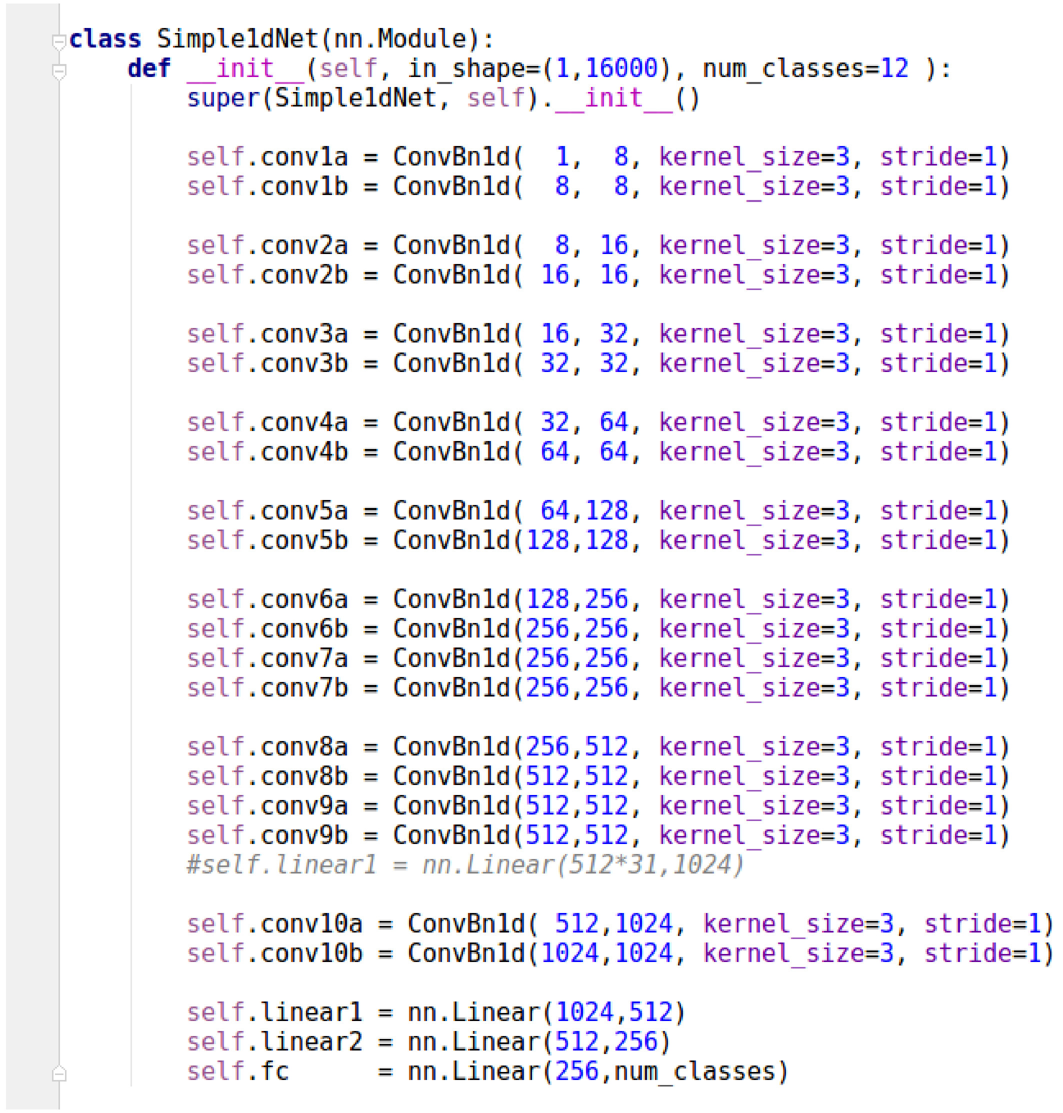
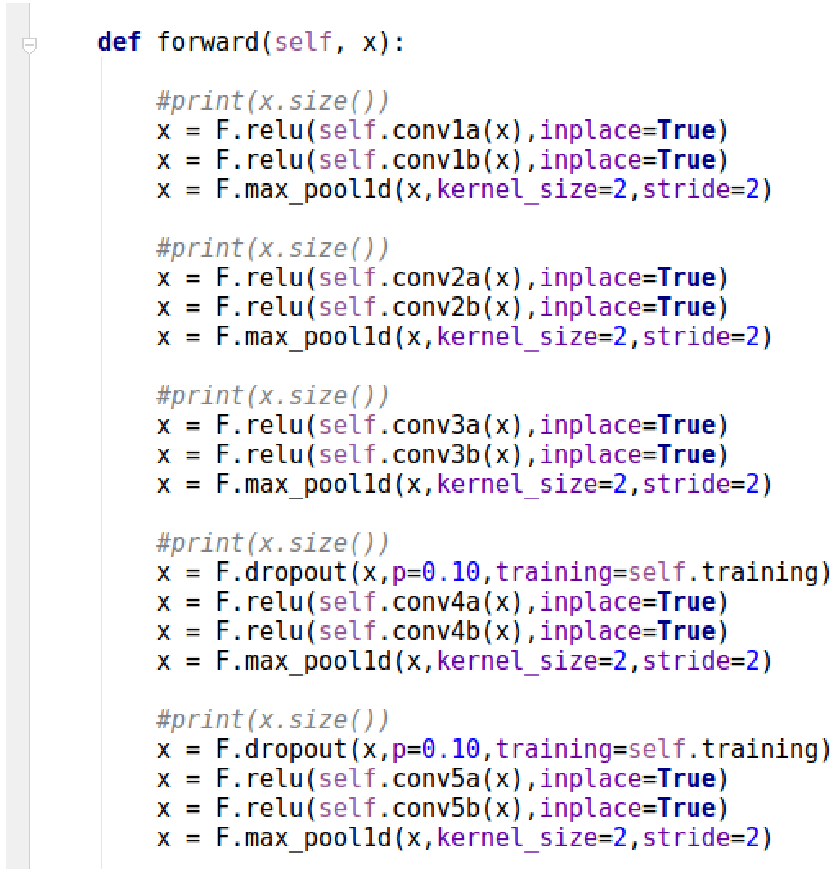
Some results for melspectrogram/ MFCC/ raw wave nets

1. raw wave (1d)

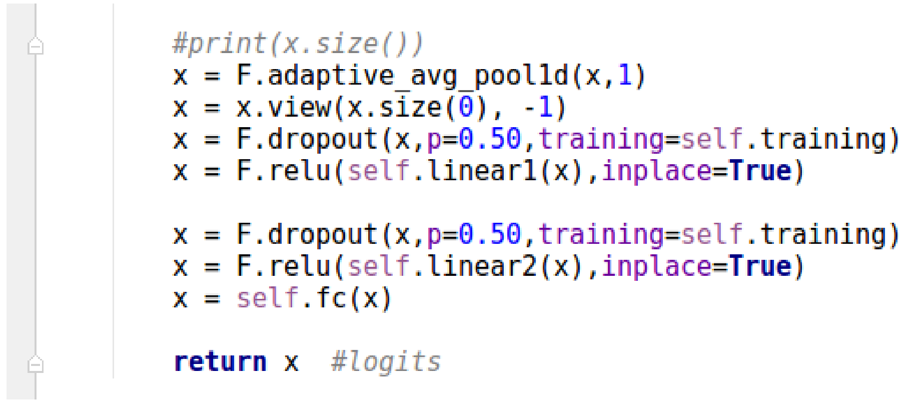


```python
class Simple1dNet(nn.Module):
    def __init__(self, in_shape=(1,16000), num_classes=12 ):
        super(Simple1dNet, self).__init__()

        self.conv1a = ConvBn1d(  1,  8, kernel_size=3, stride=1)
        self.conv1b = ConvBn1d(  8,  8, kernel_size=3, stride=1)

        self.conv2a = ConvBn1d(  8, 16, kernel_size=3, stride=1)
        self.conv2b = ConvBn1d( 16, 16, kernel_size=3, stride=1)

        self.conv3a = ConvBn1d( 16, 32, kernel_size=3, stride=1)
        self.conv3b = ConvBn1d( 32, 32, kernel_size=3, stride=1)

        self.conv4a = ConvBn1d( 32, 64, kernel_size=3, stride=1)
        self.conv4b = ConvBn1d( 64, 64, kernel_size=3, stride=1)

        self.conv5a = ConvBn1d( 64,128, kernel_size=3, stride=1)
        self.conv5b = ConvBn1d(128,128, kernel_size=3, stride=1)

        self.conv6a = ConvBn1d(128,256, kernel_size=3, stride=1)
        self.conv6b = ConvBn1d(256,256, kernel_size=3, stride=1)
        self.conv7a = ConvBn1d(256,256, kernel_size=3, stride=1)
        self.conv7b = ConvBn1d(256,256, kernel_size=3, stride=1)

        self.conv8a = ConvBn1d(256,512, kernel_size=3, stride=1)
        self.conv8b = ConvBn1d(512,512, kernel_size=3, stride=1)
        self.conv9a = ConvBn1d(512,512, kernel_size=3, stride=1)
        self.conv9b = ConvBn1d(512,512, kernel_size=3, stride=1)
        #self.linear1 = nn.Linear(512*31,1024)

        self.conv10a = ConvBn1d( 512,1024, kernel_size=3, stride=1)
        self.conv10b = ConvBn1d(1024,1024, kernel_size=3, stride=1)

        self.linear1 = nn.Linear(1024,512)
        self.linear2 = nn.Linear(512,256)
        self.fc      = nn.Linear(256,num_classes)
```



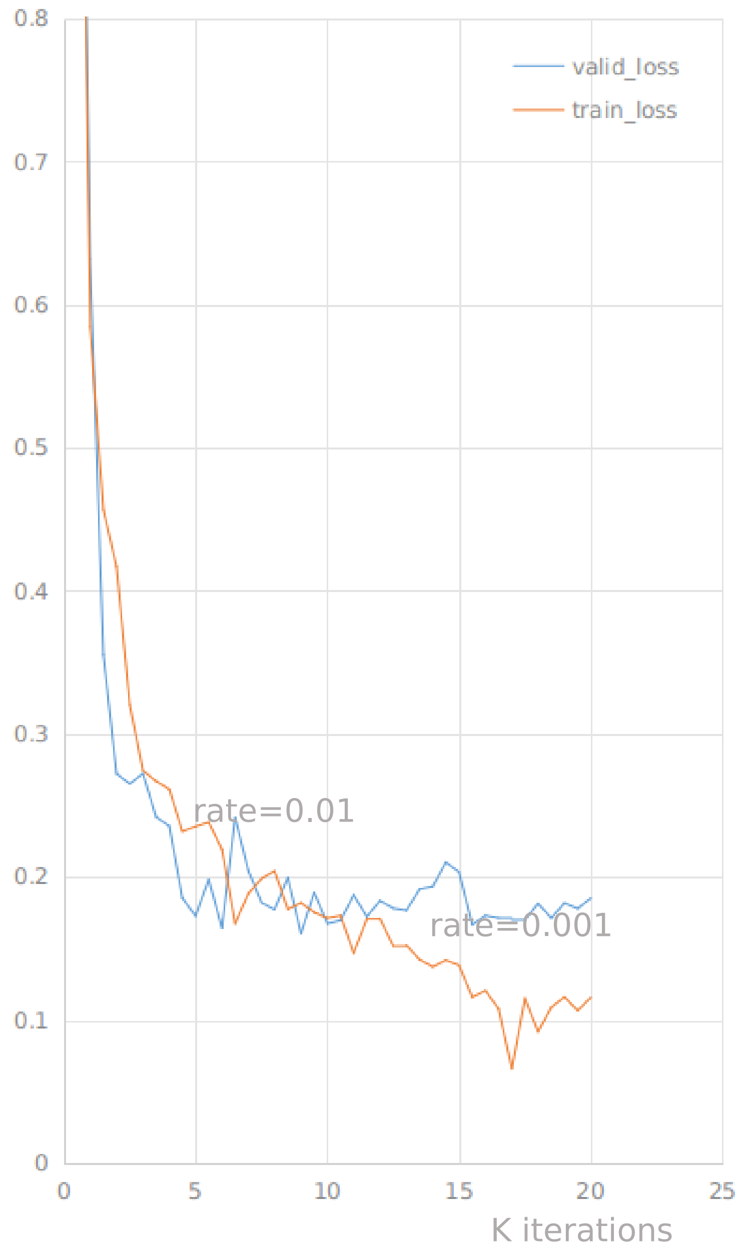
```python
    def forward(self, x):

        #print(x.size())
        x = F.relu(self.conv1a(x),inplace=True)
        x = F.relu(self.conv1b(x),inplace=True)
        x = F.max_pool1d(x,kernel_size=2,stride=2)

        #print(x.size())
        x = F.relu(self.conv2a(x),inplace=True)
        x = F.relu(self.conv2b(x),inplace=True)
        x = F.max_pool1d(x,kernel_size=2,stride=2)

        #print(x.size())
        x = F.relu(self.conv3a(x),inplace=True)
        x = F.relu(self.conv3b(x),inplace=True)
        x = F.max_pool1d(x,kernel_size=2,stride=2)

        #print(x.size())
        x = F.dropout(x,p=0.10,training=self.training)
        x = F.relu(self.conv4a(x),inplace=True)
        x = F.relu(self.conv4b(x),inplace=True)
        x = F.max_pool1d(x,kernel_size=2,stride=2)

        #print(x.size())
        x = F.dropout(x,p=0.10,training=self.training)
        x = F.relu(self.conv5a(x),inplace=True)
        x = F.relu(self.conv5b(x),inplace=True)
        x = F.max_pool1d(x,kernel_size=2,stride=2)
```

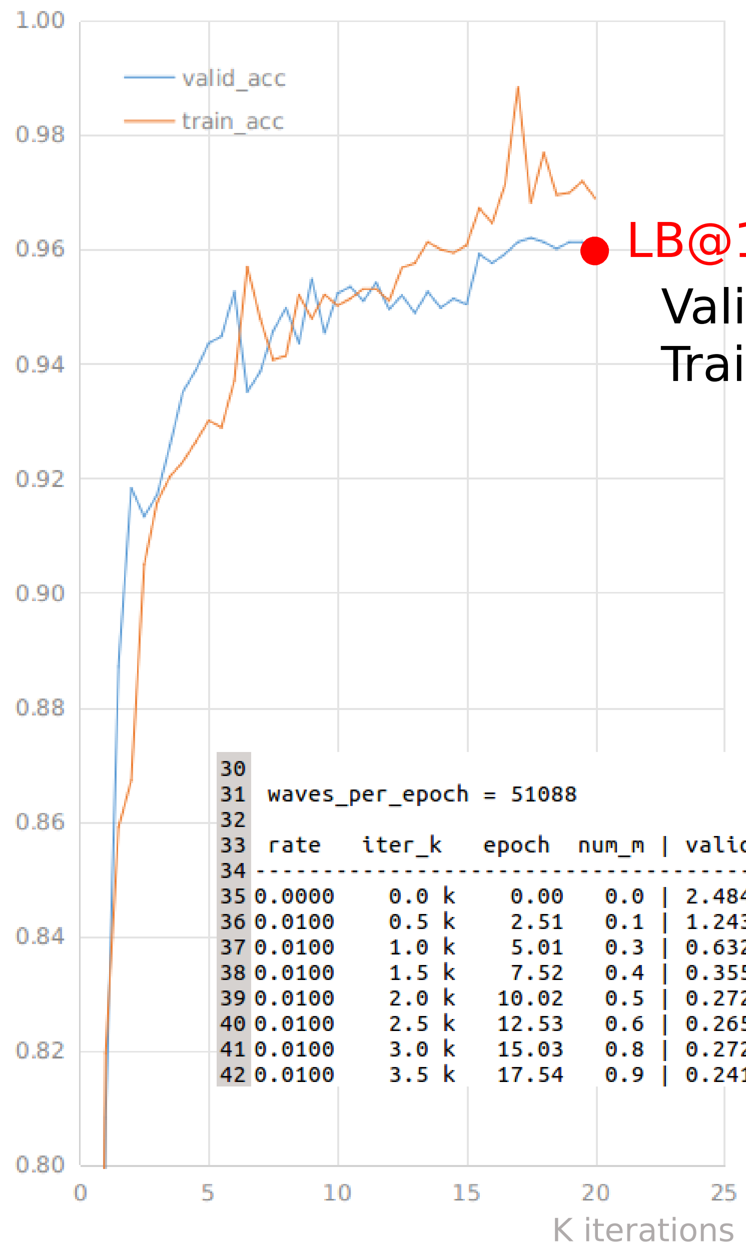
...



```python
        #print(x.size())
        x = F.adaptive_avg_pool1d(x,1)
        x = x.view(x.size(0), -1)
        x = F.dropout(x,p=0.50,training=self.training)
        x = F.relu(self.linear1(x),inplace=True)

        x = F.dropout(x,p=0.50,training=self.training)
        x = F.relu(self.linear2(x),inplace=True)
        x = self.fc(x)

        return x  #logits
```

/root/share/project/kaggle/tensorflow/results/simple1d_net-10a/checkpoint/00019000_model.pth

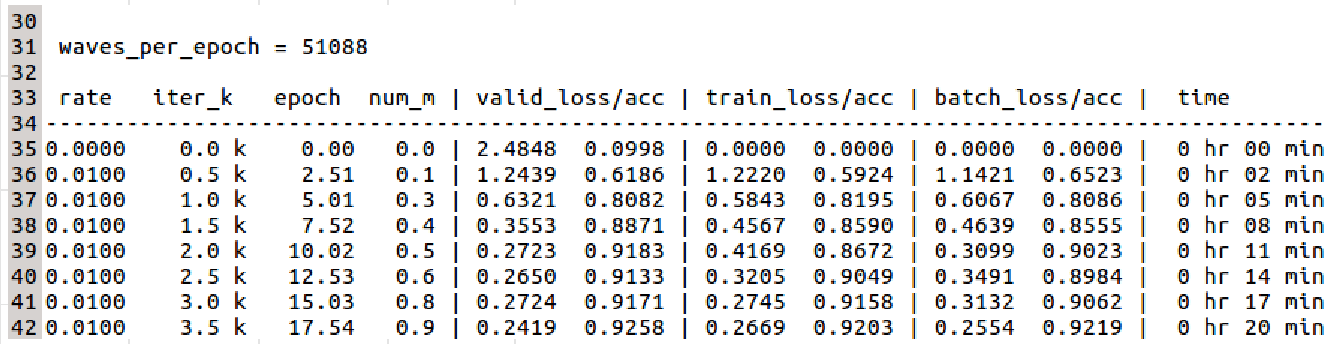




● LB@19k = 0.86

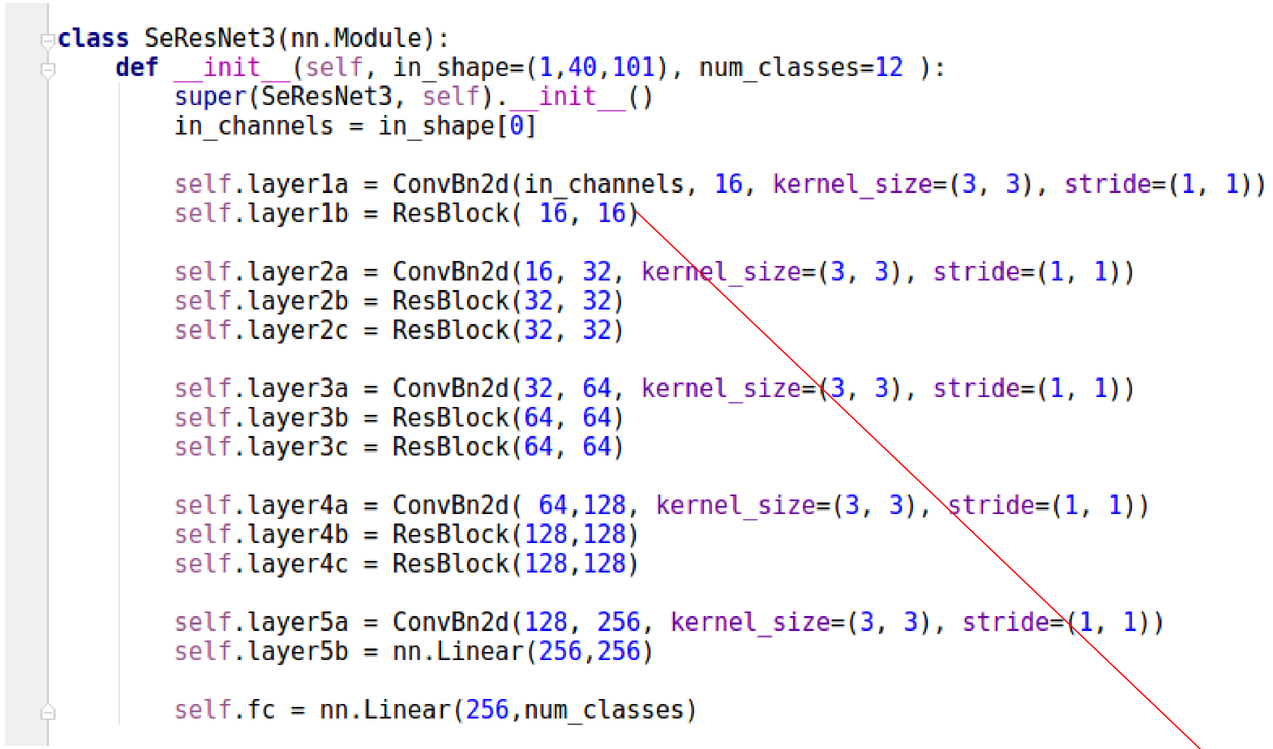
Valid = 0.1818 0.9613

Train = 0.1161 0.9699

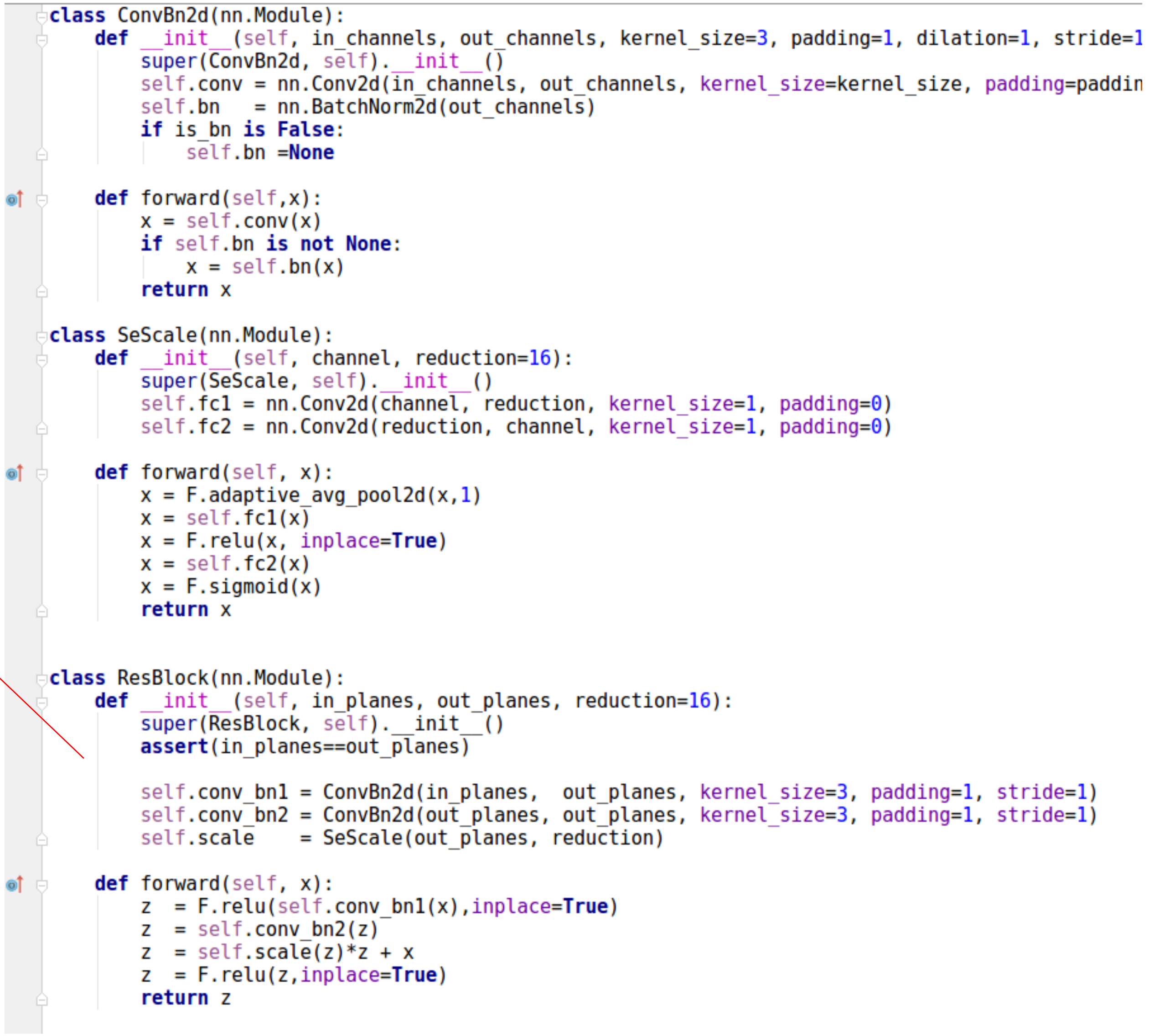


```
waves_per_epoch = 51088

rate   iter_k   epoch  num_m | valid_loss/acc | train_loss/acc | batch_loss/acc |  time
-------------------------------------------------------------------------------------------------
0.0000    0.0 k    0.00    0.0 | 2.4848  0.0998 | 0.0000  0.0000 | 0.0000  0.0000 |  0 hr 00 min
0.0100    0.5 k    2.51    0.1 | 1.2439  0.6186 | 1.2220  0.5924 | 1.1421  0.6523 |  0 hr 02 min
0.0100    1.0 k    5.01    0.3 | 0.6321  0.8082 | 0.5843  0.8195 | 0.6067  0.8086 |  0 hr 05 min
0.0100    1.5 k    7.52    0.4 | 0.3553  0.8871 | 0.4567  0.8590 | 0.4639  0.8555 |  0 hr 08 min
0.0100    2.0 k   10.02    0.5 | 0.2723  0.9183 | 0.4169  0.8672 | 0.3099  0.9023 |  0 hr 11 min
0.0100    2.5 k   12.53    0.6 | 0.2650  0.9133 | 0.3205  0.9049 | 0.3491  0.8984 |  0 hr 14 min
0.0100    3.0 k   15.03    0.8 | 0.2724  0.9171 | 0.2745  0.9158 | 0.3132  0.9062 |  0 hr 17 min
0.0100    3.5 k   17.54    0.9 | 0.2419  0.9258 | 0.2669  0.9203 | 0.2554  0.9219 |  0 hr 20 min
```

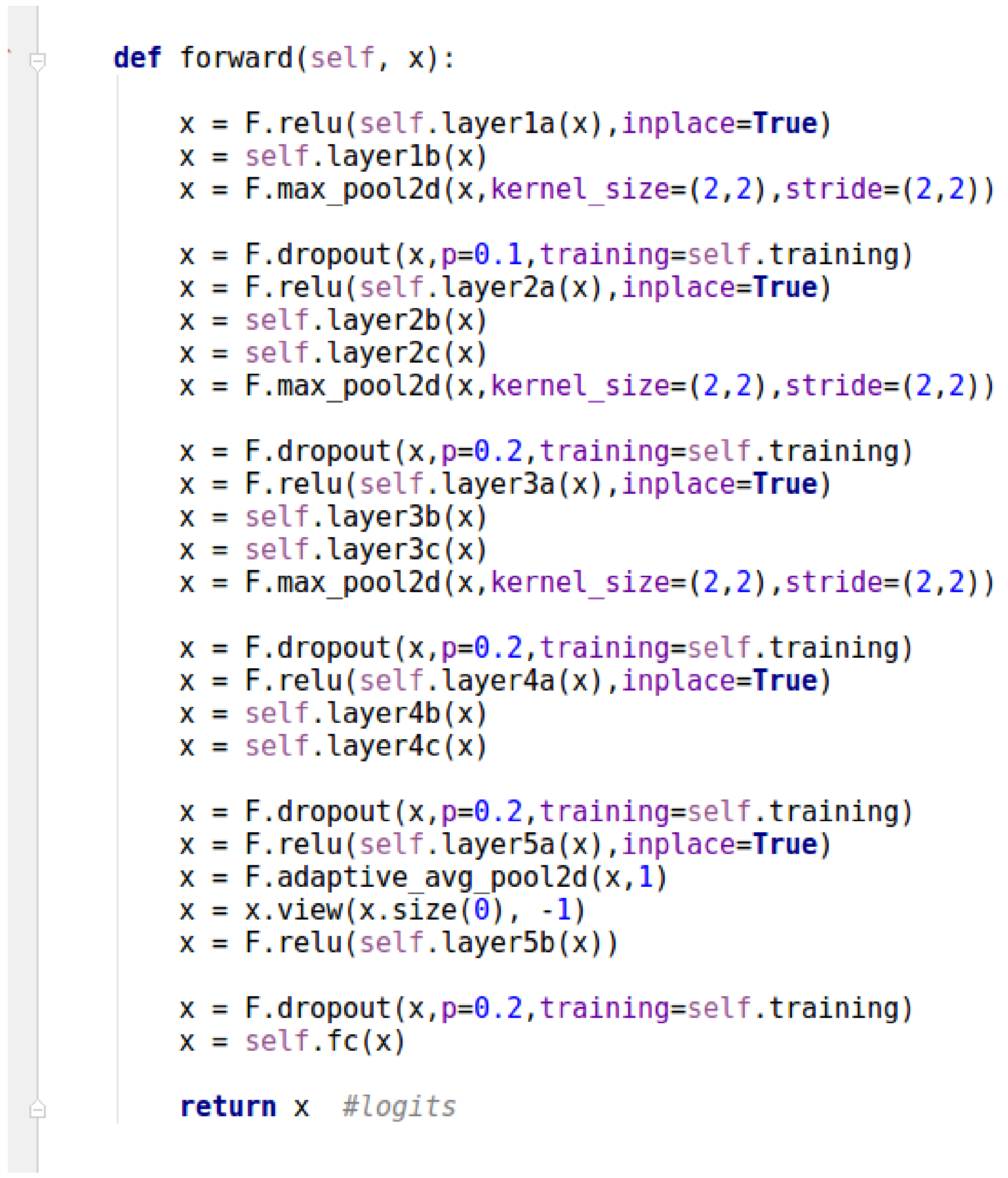
2. melspectrogram (2d)



```python
class SeResNet3(nn.Module):
    def __init__(self, in_shape=(1,40,101), num_classes=12 ):
        super(SeResNet3, self).__init__()
        in_channels = in_shape[0]

        self.layer1a = ConvBn2d(in_channels, 16, kernel_size=(3, 3), stride=(1, 1))
        self.layer1b = ResBlock( 16, 16)

        self.layer2a = ConvBn2d(16, 32, kernel_size=(3, 3), stride=(1, 1))
        self.layer2b = ResBlock(32, 32)
        self.layer2c = ResBlock(32, 32)

        self.layer3a = ConvBn2d(32, 64, kernel_size=(3, 3), stride=(1, 1))
        self.layer3b = ResBlock(64, 64)
        self.layer3c = ResBlock(64, 64)

        self.layer4a = ConvBn2d( 64,128, kernel_size=(3, 3), stride=(1, 1))
        self.layer4b = ResBlock(128,128)
        self.layer4c = ResBlock(128,128)

        self.layer5a = ConvBn2d(128, 256, kernel_size=(3, 3), stride=(1, 1))
        self.layer5b = nn.Linear(256,256)

        self.fc = nn.Linear(256,num_classes)
```

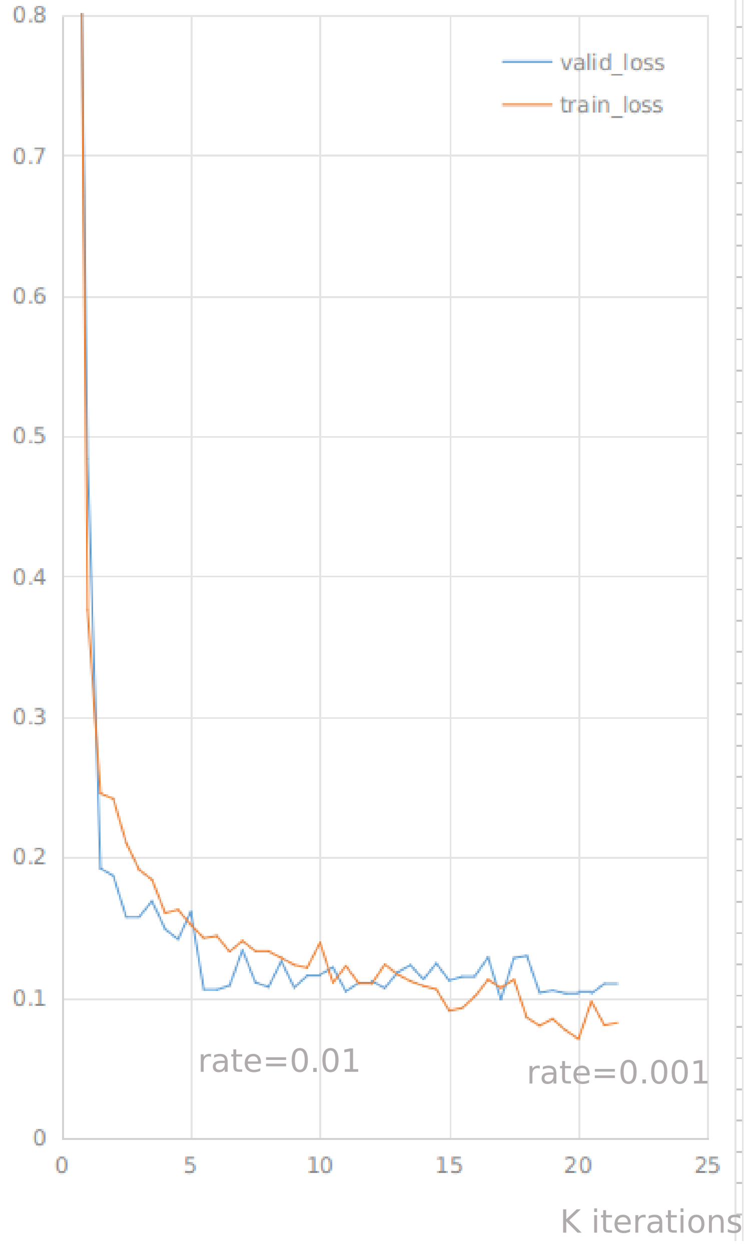


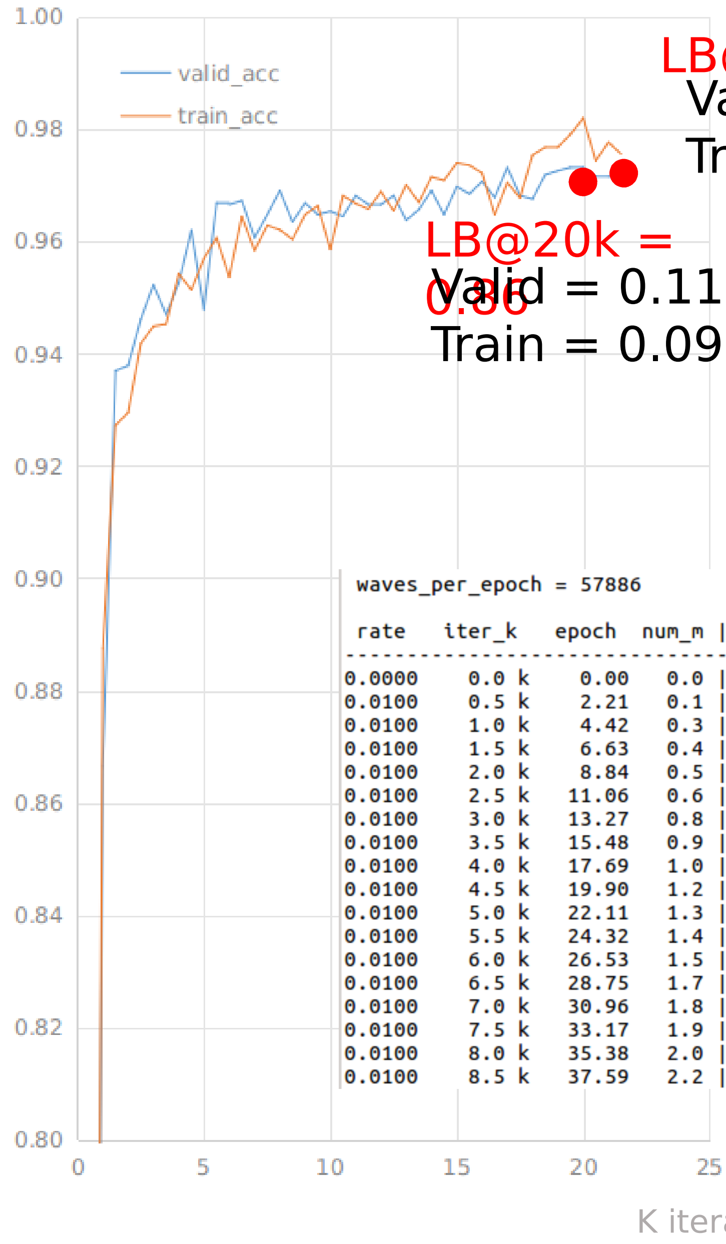
```python
class ConvBn2d(nn.Module):
    def __init__(self, in_channels, out_channels, kernel_size=3, padding=1, dilation=1, stride=1
        super(ConvBn2d, self).__init__()
        self.conv = nn.Conv2d(in_channels, out_channels, kernel_size=kernel_size, padding=paddin
        self.bn   = nn.BatchNorm2d(out_channels)
        if is_bn is False:
            self.bn =None

    def forward(self,x):
        x = self.conv(x)
        if self.bn is not None:
            x = self.bn(x)
        return x

class SeScale(nn.Module):
    def __init__(self, channel, reduction=16):
        super(SeScale, self).__init__()
        self.fc1 = nn.Conv2d(channel, reduction, kernel_size=1, padding=0)
        self.fc2 = nn.Conv2d(reduction, channel, kernel_size=1, padding=0)

    def forward(self, x):
        x = F.adaptive_avg_pool2d(x,1)
        x = self.fc1(x)
        x = F.relu(x, inplace=True)
        x = self.fc2(x)
        x = F.sigmoid(x)
        return x


class ResBlock(nn.Module):
    def __init__(self, in_planes, out_planes, reduction=16):
        super(ResBlock, self).__init__()
        assert(in_planes==out_planes)

        self.conv_bn1 = ConvBn2d(in_planes,  out_planes, kernel_size=3, padding=1, stride=1)
        self.conv_bn2 = ConvBn2d(out_planes, out_planes, kernel_size=3, padding=1, stride=1)
        self.scale    = SeScale(out_planes, reduction)

    def forward(self, x):
        z  = F.relu(self.conv_bn1(x),inplace=True)
        z  = self.conv_bn2(z)
        z  = self.scale(z)*z + x
        z  = F.relu(z,inplace=True)
        return z
```



```
def forward(self, x):

    x = F.relu(self.layer1a(x),inplace=True)
    x = self.layer1b(x)
    x = F.max_pool2d(x,kernel_size=(2,2),stride=(2,2))

    x = F.dropout(x,p=0.1,training=self.training)
    x = F.relu(self.layer2a(x),inplace=True)
    x = self.layer2b(x)
    x = self.layer2c(x)
    x = F.max_pool2d(x,kernel_size=(2,2),stride=(2,2))

    x = F.dropout(x,p=0.2,training=self.training)
    x = F.relu(self.layer3a(x),inplace=True)
    x = self.layer3b(x)
    x = self.layer3c(x)
    x = F.max_pool2d(x,kernel_size=(2,2),stride=(2,2))

    x = F.dropout(x,p=0.2,training=self.training)
    x = F.relu(self.layer4a(x),inplace=True)
    x = self.layer4b(x)
    x = self.layer4c(x)

    x = F.dropout(x,p=0.2,training=self.training)
    x = F.relu(self.layer5a(x),inplace=True)
    x = F.adaptive_avg_pool2d(x,1)
    x = x.view(x.size(0), -1)
    x = F.relu(self.layer5b(x))

    x = F.dropout(x,p=0.2,training=self.training)
    x = self.fc(x)

    return x  #logits
```

/root/share/project/kaggle/tensorflow/results/se-resnet3-spect-02a/checkpoint/00021000_model.pth





LB@21k =?

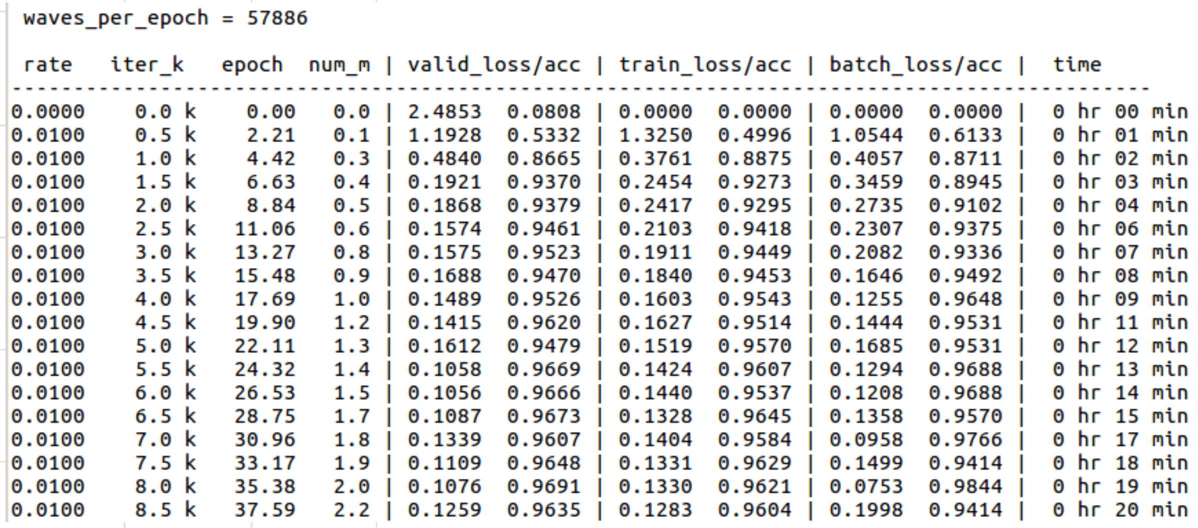
Valid = 0.1099 0.9716

Train = 0.0804 0.9777

LB@20k = 0.86

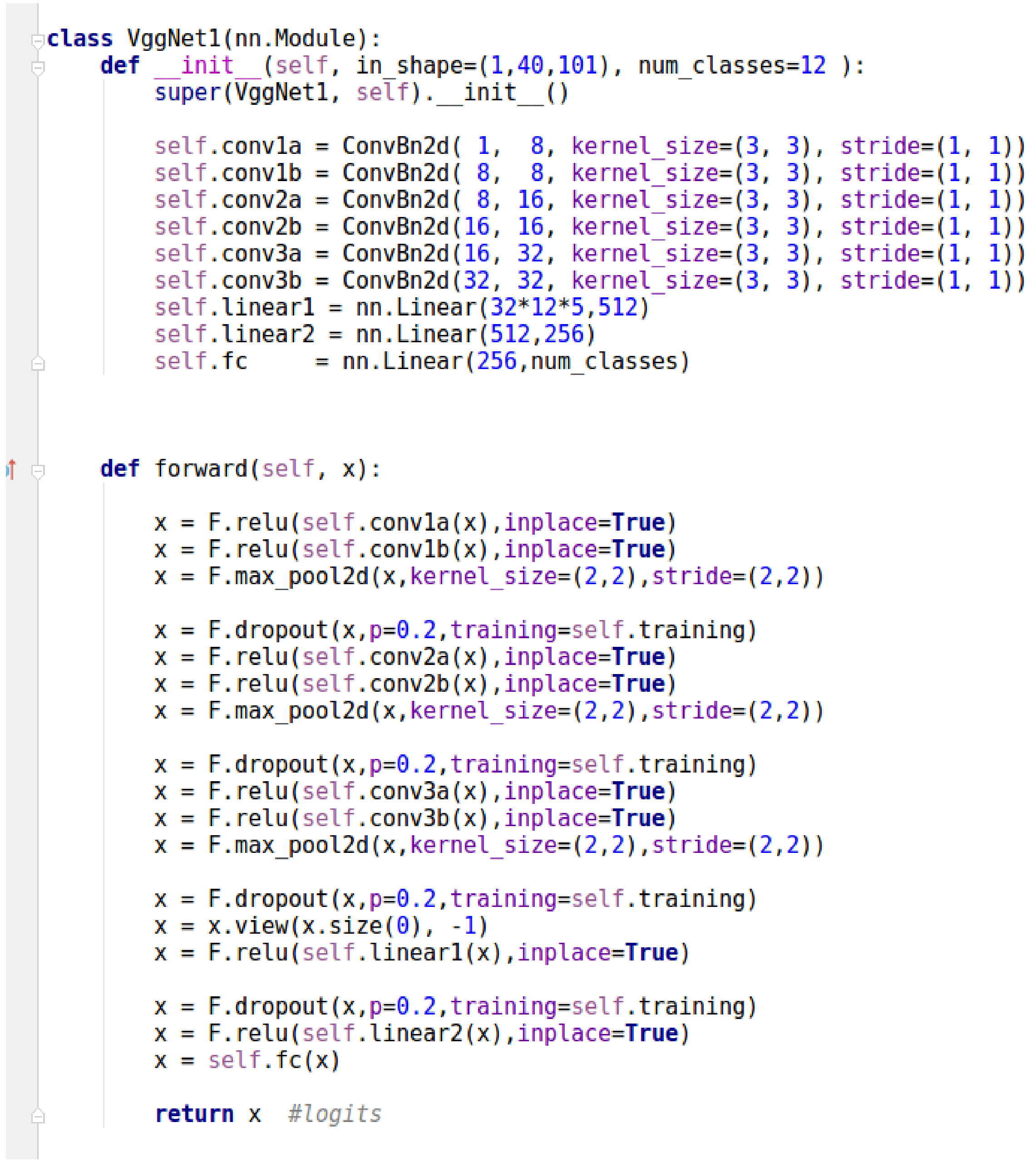
Valid = 0.1130 0.9688

Train = 0.0988 0.9719



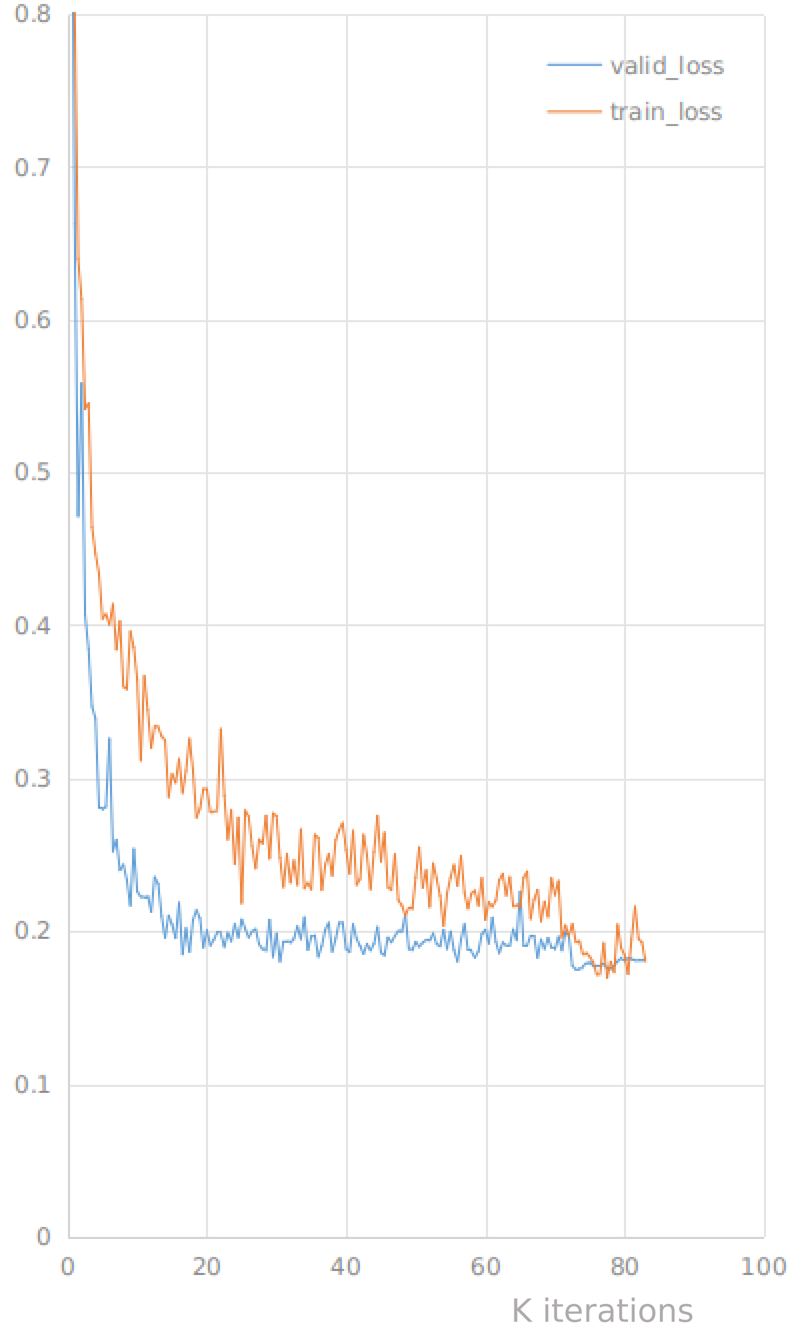
```
 waves_per_epoch = 57886

 rate    iter_k   epoch  num_m | valid_loss/acc | train_loss/acc | batch_loss/acc |  time
--------------------------------------------------------------------------------------------
0.0000    0.0 k    0.00   0.0 | 2.4853  0.0808 | 0.0000  0.0000 | 0.0000  0.0000 |  0 hr 00 min
0.0100    0.5 k    2.21   0.1 | 1.1928  0.5332 | 1.3250  0.4996 | 1.0544  0.6133 |  0 hr 01 min
0.0100    1.0 k    4.42   0.3 | 0.4840  0.8665 | 0.3761  0.8875 | 0.4057  0.8711 |  0 hr 02 min
0.0100    1.5 k    6.63   0.4 | 0.1921  0.9370 | 0.2454  0.9273 | 0.3459  0.8945 |  0 hr 03 min
0.0100    2.0 k    8.84   0.5 | 0.1868  0.9379 | 0.2417  0.9295 | 0.2735  0.9102 |  0 hr 04 min
0.0100    2.5 k   11.06   0.6 | 0.1574  0.9461 | 0.2103  0.9418 | 0.2307  0.9375 |  0 hr 06 min
0.0100    3.0 k   13.27   0.8 | 0.1575  0.9523 | 0.1911  0.9449 | 0.2082  0.9336 |  0 hr 07 min
0.0100    3.5 k   15.48   0.9 | 0.1688  0.9470 | 0.1840  0.9453 | 0.1646  0.9492 |  0 hr 08 min
0.0100    4.0 k   17.69   1.0 | 0.1489  0.9526 | 0.1603  0.9543 | 0.1255  0.9648 |  0 hr 09 min
0.0100    4.5 k   19.90   1.2 | 0.1415  0.9620 | 0.1627  0.9514 | 0.1444  0.9531 |  0 hr 11 min
0.0100    5.0 k   22.11   1.3 | 0.1612  0.9479 | 0.1519  0.9570 | 0.1685  0.9531 |  0 hr 12 min
0.0100    5.5 k   24.32   1.4 | 0.1058  0.9669 | 0.1424  0.9607 | 0.1294  0.9688 |  0 hr 13 min
0.0100    6.0 k   26.53   1.5 | 0.1056  0.9666 | 0.1440  0.9537 | 0.1208  0.9688 |  0 hr 14 min
0.0100    6.5 k   28.75   1.7 | 0.1087  0.9673 | 0.1328  0.9645 | 0.1358  0.9570 |  0 hr 15 min
0.0100    7.0 k   30.96   1.8 | 0.1339  0.9607 | 0.1404  0.9584 | 0.0958  0.9766 |  0 hr 17 min
0.0100    7.5 k   33.17   1.9 | 0.1109  0.9648 | 0.1331  0.9629 | 0.1499  0.9414 |  0 hr 18 min
0.0100    8.0 k   35.38   2.0 | 0.1076  0.9691 | 0.1330  0.9621 | 0.0753  0.9844 |  0 hr 19 min
0.0100    8.5 k   37.59   2.2 | 0.1259  0.9635 | 0.1283  0.9604 | 0.1998  0.9414 |  0 hr 20 min
```

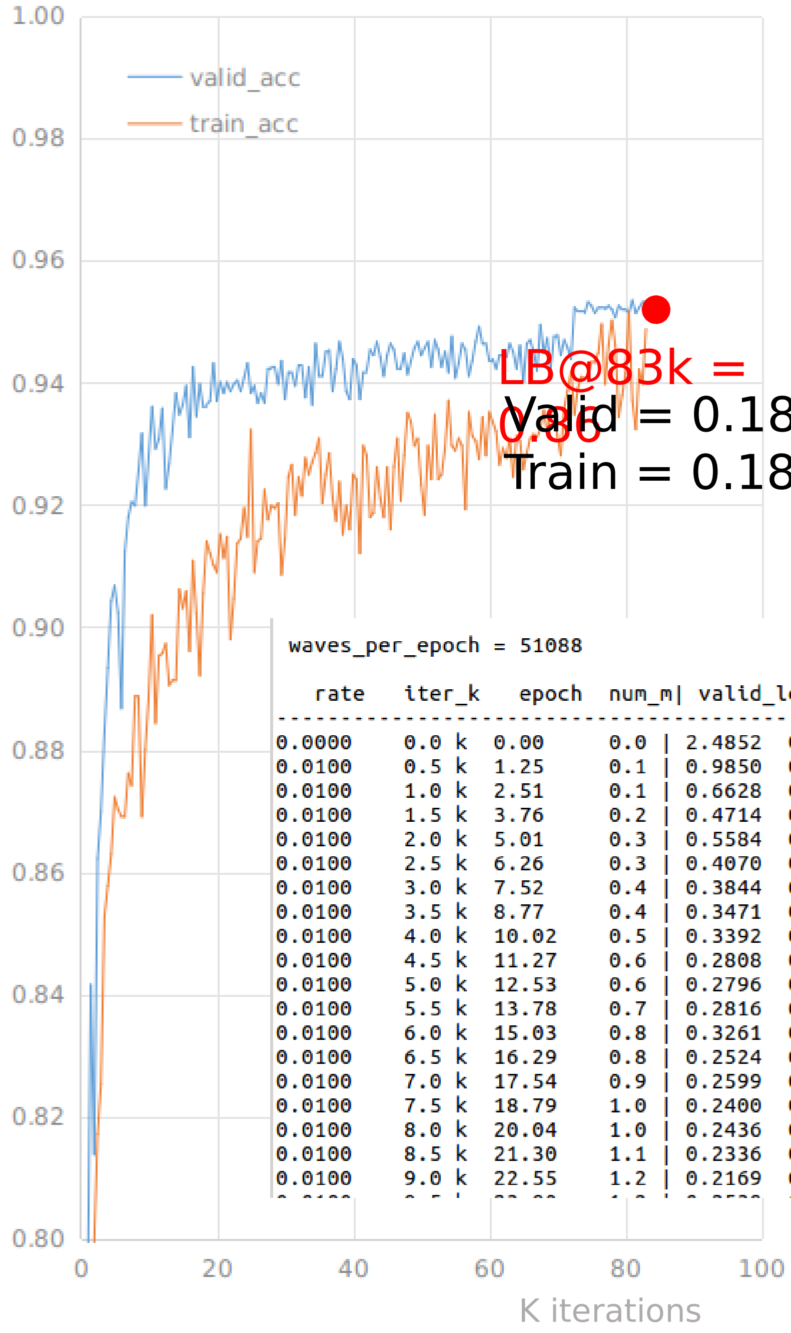
3. mfcc(2d)



```
class VggNet1(nn.Module):
    def __init__(self, in_shape=(1,40,101), num_classes=12 ):
        super(VggNet1, self).__init__()

        self.conv1a = ConvBn2d( 1,  8, kernel_size=(3, 3), stride=(1, 1))
        self.conv1b = ConvBn2d( 8,  8, kernel_size=(3, 3), stride=(1, 1))
        self.conv2a = ConvBn2d( 8, 16, kernel_size=(3, 3), stride=(1, 1))
        self.conv2b = ConvBn2d(16, 16, kernel_size=(3, 3), stride=(1, 1))
        self.conv3a = ConvBn2d(16, 32, kernel_size=(3, 3), stride=(1, 1))
        self.conv3b = ConvBn2d(32, 32, kernel_size=(3, 3), stride=(1, 1))
        self.linear1 = nn.Linear(32*12*5,512)
        self.linear2 = nn.Linear(512,256)
        self.fc     = nn.Linear(256,num_classes)



    def forward(self, x):

        x = F.relu(self.conv1a(x),inplace=True)
        x = F.relu(self.conv1b(x),inplace=True)
        x = F.max_pool2d(x,kernel_size=(2,2),stride=(2,2))

        x = F.dropout(x,p=0.2,training=self.training)
        x = F.relu(self.conv2a(x),inplace=True)
        x = F.relu(self.conv2b(x),inplace=True)
        x = F.max_pool2d(x,kernel_size=(2,2),stride=(2,2))

        x = F.dropout(x,p=0.2,training=self.training)
        x = F.relu(self.conv3a(x),inplace=True)
        x = F.relu(self.conv3b(x),inplace=True)
        x = F.max_pool2d(x,kernel_size=(2,2),stride=(2,2))

        x = F.dropout(x,p=0.2,training=self.training)
        x = x.view(x.size(0), -1)
        x = F.relu(self.linear1(x),inplace=True)

        x = F.dropout(x,p=0.2,training=self.training)
        x = F.relu(self.linear2(x),inplace=True)
        x = self.fc(x)

        return x  #logits
```

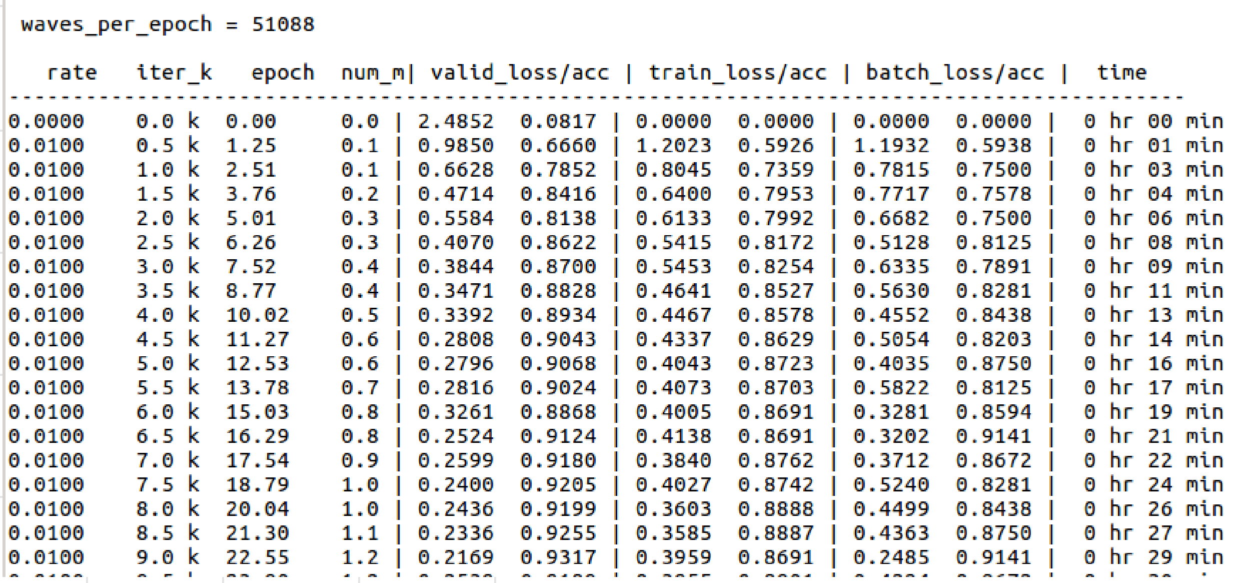
/root/share/project/kaggle/tensorflow/results/vgg1_net-06b/checkpoint/00083000_model.pth





LB@83k = 0.86

Valid = 0.1821 0.9532
Train = 0.1803 0.9488



```
waves_per_epoch = 51088

  rate    iter_k   epoch   num_m| valid_loss/acc | train_loss/acc | batch_loss/acc |   time
------------------------------------------------------------------------------------------------
0.0000    0.0 k   0.00      0.0 | 2.4852  0.0817 | 0.0000  0.0000 | 0.0000  0.0000 |  0 hr 00 min
0.0100    0.5 k   1.25      0.1 | 0.9850  0.6660 | 1.2023  0.5926 | 1.1932  0.5938 |  0 hr 01 min
0.0100    1.0 k   2.51      0.1 | 0.6628  0.7852 | 0.8045  0.7359 | 0.7815  0.7500 |  0 hr 03 min
0.0100    1.5 k   3.76      0.2 | 0.4714  0.8416 | 0.6400  0.7953 | 0.7717  0.7578 |  0 hr 04 min
0.0100    2.0 k   5.01      0.3 | 0.5584  0.8138 | 0.6133  0.7992 | 0.6682  0.7500 |  0 hr 06 min
0.0100    2.5 k   6.26      0.3 | 0.4070  0.8622 | 0.5415  0.8172 | 0.5128  0.8125 |  0 hr 08 min
0.0100    3.0 k   7.52      0.4 | 0.3844  0.8700 | 0.5453  0.8254 | 0.6335  0.7891 |  0 hr 09 min
0.0100    3.5 k   8.77      0.4 | 0.3471  0.8828 | 0.4641  0.8527 | 0.5630  0.8281 |  0 hr 11 min
0.0100    4.0 k   10.02     0.5 | 0.3392  0.8934 | 0.4467  0.8578 | 0.4552  0.8438 |  0 hr 13 min
0.0100    4.5 k   11.27     0.6 | 0.2808  0.9043 | 0.4337  0.8629 | 0.5054  0.8203 |  0 hr 14 min
0.0100    5.0 k   12.53     0.6 | 0.2796  0.9068 | 0.4043  0.8723 | 0.4035  0.8750 |  0 hr 16 min
0.0100    5.5 k   13.78     0.7 | 0.2816  0.9024 | 0.4073  0.8703 | 0.5822  0.8125 |  0 hr 17 min
0.0100    6.0 k   15.03     0.8 | 0.3261  0.8868 | 0.4005  0.8691 | 0.3281  0.8594 |  0 hr 19 min
0.0100    6.5 k   16.29     0.8 | 0.2524  0.9124 | 0.4138  0.8691 | 0.3202  0.9141 |  0 hr 21 min
0.0100    7.0 k   17.54     0.9 | 0.2599  0.9180 | 0.3840  0.8762 | 0.3712  0.8672 |  0 hr 22 min
0.0100    7.5 k   18.79     1.0 | 0.2400  0.9205 | 0.4027  0.8742 | 0.5240  0.8281 |  0 hr 24 min
0.0100    8.0 k   20.04     1.0 | 0.2436  0.9199 | 0.3603  0.8888 | 0.4499  0.8438 |  0 hr 26 min
0.0100    8.5 k   21.30     1.1 | 0.2336  0.9255 | 0.3585  0.8887 | 0.4363  0.8750 |  0 hr 27 min
0.0100    9.0 k   22.55     1.2 | 0.2169  0.9317 | 0.3959  0.8691 | 0.2485  0.9141 |  0 hr 29 min
```