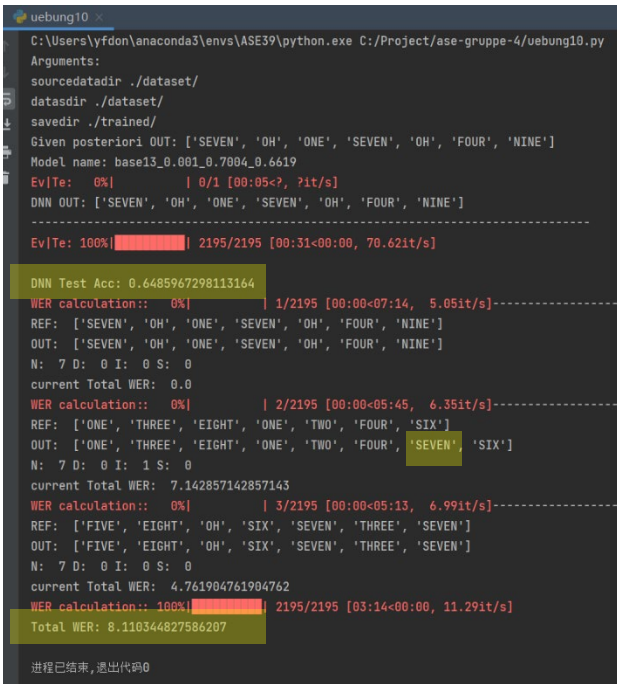
Aufgabe 11

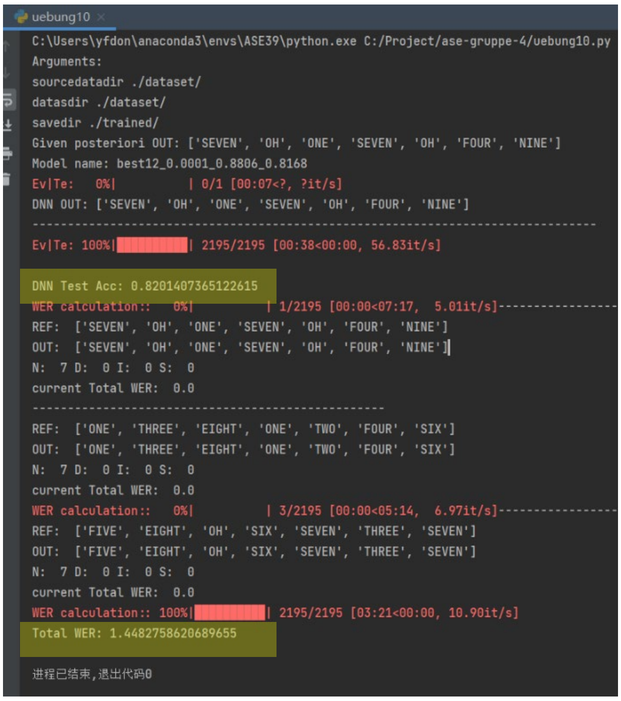
Gruppe 4

Modell	# DNN-Parameter (Epoch_lr_Train_Evaluation)	DNN-Accuracy(Test)	Test-WER
Baseline	**<u>13_0.001_0.7004_0.6619</u>** "lr": 0.001, "batch_size": 1, "epochs": 50, "window_size": 25e-3, "hop_size": 10e-3, "feature_type": "MFCC_D_DD", "n_filters": 40, "fbank_fmin": 0, "fbank_fmax": 8000, "num_ceps": 13, "left_context": 10, "right_context": 10,	0.6485967298113164	8.110344827586207
Best	**<u>9_0.000001_0.8392_0.7920</u>** *Siehe nächste Seite*	0.8201407365122615	1.4482758620689655



```
C:\Users\yfdon\anaconda3\envs\ASE39\python.exe C:/Project/ase-gruppe-4/uebung10.py
Arguments:
sourcedatadir ./dataset/
datasdir ./dataset/
savedir ./trained/
Given posteriori OUT: ['SEVEN', 'OH', 'ONE', 'SEVEN', 'OH', 'FOUR', 'NINE']
Model name: base13_0.001_0.7004_0.6619
Ev|Te:   0%|          | 0/1 [00:05<?, ?it/s]
DNN OUT: ['SEVEN', 'OH', 'ONE', 'SEVEN', 'OH', 'FOUR', 'NINE']
--------------------------------------------------------------------------------
Ev|Te: 100%|██████████| 2195/2195 [00:31<00:00, 70.62it/s]

DNN Test Acc: 0.6485967298113164
WER calculation::   0%|          | 1/2195 [00:00<07:14,  5.05it/s]------------------
REF:  ['SEVEN', 'OH', 'ONE', 'SEVEN', 'OH', 'FOUR', 'NINE']
OUT:  ['SEVEN', 'OH', 'ONE', 'SEVEN', 'OH', 'FOUR', 'NINE']
N:  7 D:  0 I:  0 S:  0
current Total WER:  0.0
WER calculation::   0%|          | 2/2195 [00:00<05:45,  6.35it/s]------------------
REF:  ['ONE', 'THREE', 'EIGHT', 'ONE', 'TWO', 'FOUR', 'SIX']
OUT:  ['ONE', 'THREE', 'EIGHT', 'ONE', 'TWO', 'FOUR', 'SEVEN', 'SIX']
N:  7 D:  0 I:  1 S:  0
current Total WER:  7.142857142857143
WER calculation::   0%|          | 3/2195 [00:00<05:13,  6.99it/s]------------------
REF:  ['FIVE', 'EIGHT', 'OH', 'SIX', 'SEVEN', 'THREE', 'SEVEN']
OUT:  ['FIVE', 'EIGHT', 'OH', 'SIX', 'SEVEN', 'THREE', 'SEVEN']
N:  7 D:  0 I:  0 S:  0
current Total WER:  4.761904761904762
WER calculation:: 100%|██████████| 2195/2195 [03:14<00:00, 11.29it/s]
Total WER: 8.110344827586207

进程已结束,退出代码0
```

Baseline



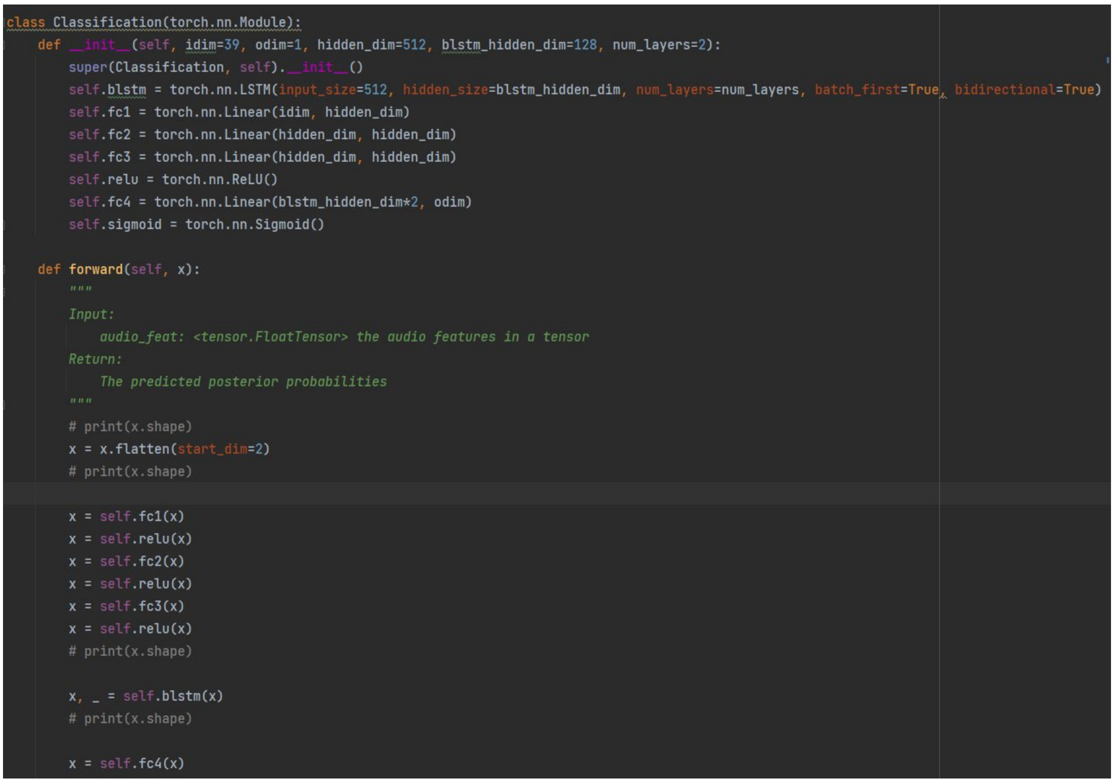
```
C:\Users\yfdon\anaconda3\envs\ASE39\python.exe C:/Project/ase-gruppe-4/uebung10.py
Arguments:
sourcedatadir ./dataset/
datasdir ./dataset/
savedir ./trained/
Given posteriori OUT: ['SEVEN', 'OH', 'ONE', 'SEVEN', 'OH', 'FOUR', 'NINE']
Model name: best12_0.0001_0.8806_0.8168
Ev|Te:   0%|          | 0/1 [00:07<?, ?it/s]
DNN OUT: ['SEVEN', 'OH', 'ONE', 'SEVEN', 'OH', 'FOUR', 'NINE']
--------------------------------------------------------------------------------
Ev|Te: 100%|██████████| 2195/2195 [00:38<00:00, 56.83it/s]

DNN Test Acc: 0.8201407365122615
WER calculation::   0%|          | 1/2195 [00:00<07:17,  5.01it/s]----------------
REF:  ['SEVEN', 'OH', 'ONE', 'SEVEN', 'OH', 'FOUR', 'NINE']
OUT:  ['SEVEN', 'OH', 'ONE', 'SEVEN', 'OH', 'FOUR', 'NINE']
N:  7 D:  0 I:  0 S:  0
current Total WER:  0.0
--------------------------------------------------
REF:  ['ONE', 'THREE', 'EIGHT', 'ONE', 'TWO', 'FOUR', 'SIX']
OUT:  ['ONE', 'THREE', 'EIGHT', 'ONE', 'TWO', 'FOUR', 'SIX']
N:  7 D:  0 I:  0 S:  0
current Total WER:  0.0
WER calculation::   0%|          | 3/2195 [00:00<05:14,  6.97it/s]----------------
REF:  ['FIVE', 'EIGHT', 'OH', 'SIX', 'SEVEN', 'THREE', 'SEVEN']
OUT:  ['FIVE', 'EIGHT', 'OH', 'SIX', 'SEVEN', 'THREE', 'SEVEN']
N:  7 D:  0 I:  0 S:  0
current Total WER:  0.0
WER calculation:: 100%|██████████| 2195/2195 [03:21<00:00, 10.90it/s]
Total WER: 1.4482758620689655

进程已结束,退出代码0
```

Best

Best:

1. Layer: BLSTM Layer nach den FC-Layers
2. Learning Rate: ein StepLR scheduler wurde implementiert.

Verwenden Sie **train2.py** und **model2.py**, um dieses verbesserte Modell zu trainieren.



```python
class Classification(torch.nn.Module):
    def __init__(self, idim=39, odim=1, hidden_dim=512, blstm_hidden_dim=128, num_layers=2):
        super(Classification, self).__init__()
        self.blstm = torch.nn.LSTM(input_size=512, hidden_size=blstm_hidden_dim, num_layers=num_layers, batch_first=True, bidirectional=True)
        self.fc1 = torch.nn.Linear(idim, hidden_dim)
        self.fc2 = torch.nn.Linear(hidden_dim, hidden_dim)
        self.fc3 = torch.nn.Linear(hidden_dim, hidden_dim)
        self.relu = torch.nn.ReLU()
        self.fc4 = torch.nn.Linear(blstm_hidden_dim*2, odim)
        self.sigmoid = torch.nn.Sigmoid()

    def forward(self, x):
        """
        Input:
            audio_feat: <tensor.FloatTensor> the audio features in a tensor
        Return:
            The predicted posterior probabilities
        """
        # print(x.shape)
        x = x.flatten(start_dim=2)
        # print(x.shape)

        x = self.fc1(x)
        x = self.relu(x)
        x = self.fc2(x)
        x = self.relu(x)
        x = self.fc3(x)
        x = self.relu(x)
        # print(x.shape)

        x, _ = self.blstm(x)
        # print(x.shape)

        x = self.fc4(x)
```