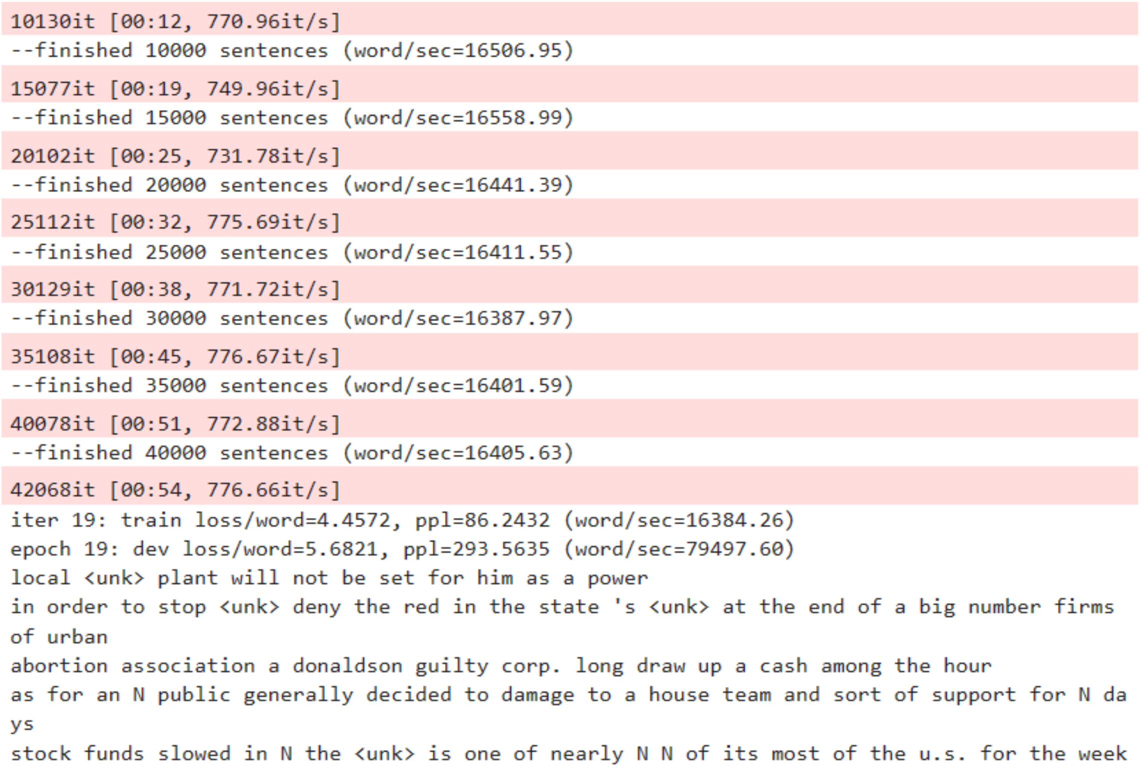
语音实验报告

高祎珂2011743

运行结果截图

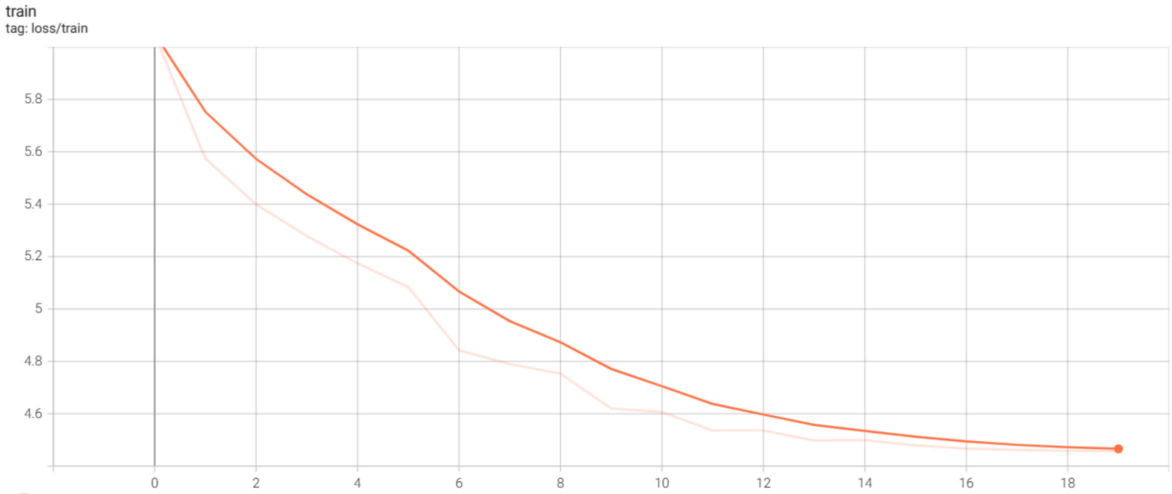
根据给出的函数，使用云服务器，设置epoch为20，进行模型训练可以得到结果如下图所示：



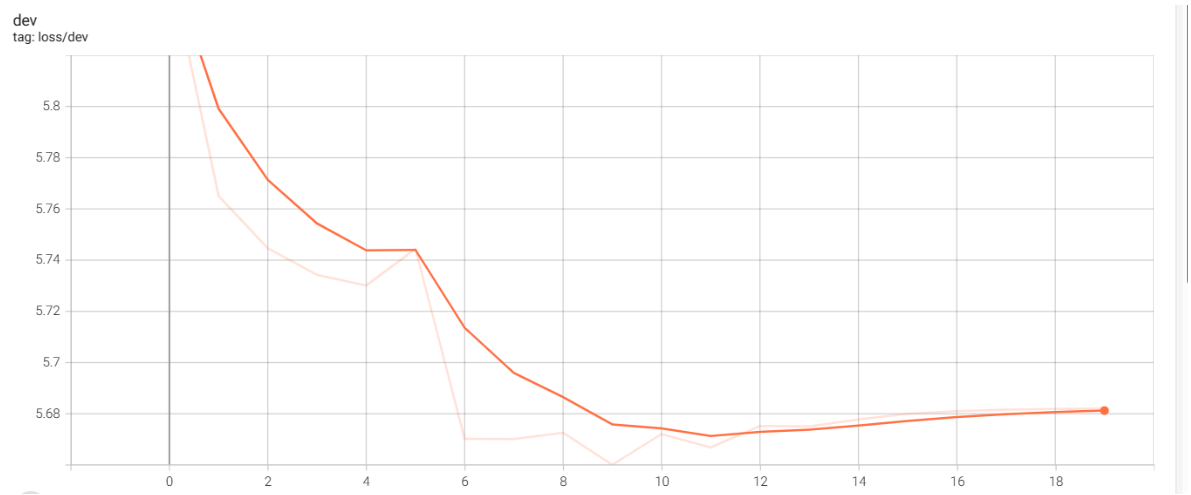
```
10130it [00:12, 770.96it/s]
--finished 10000 sentences (word/sec=16506.95)
15077it [00:19, 749.96it/s]
--finished 15000 sentences (word/sec=16558.99)
20102it [00:25, 731.78it/s]
--finished 20000 sentences (word/sec=16441.39)
25112it [00:32, 775.69it/s]
--finished 25000 sentences (word/sec=16411.55)
30129it [00:38, 771.72it/s]
--finished 30000 sentences (word/sec=16387.97)
35108it [00:45, 776.67it/s]
--finished 35000 sentences (word/sec=16401.59)
40078it [00:51, 772.88it/s]
--finished 40000 sentences (word/sec=16405.63)
42068it [00:54, 776.66it/s]
iter 19: train loss/word=4.4572, ppl=86.2432 (word/sec=16384.26)
epoch 19: dev loss/word=5.6821, ppl=293.5635 (word/sec=79497.60)
local <unk> plant will not be set for him as a power
in order to stop <unk> deny the red in the state 's <unk> at the end of a big number firms
of urban
abortion association a donaldson guilty corp. long draw up a cash among the hour
as for an N public generally decided to damage to a house team and sort of support for N da
ys
stock funds slowed in N the <unk> is one of nearly N N of its most of the u.s. for the week
```

训练集和验证集上的loss曲线

设置epoch为20，绘制训练集和验证集的loss图像如下图所示。从图中我们可以看出，在训练过程中，训练集和验证集的loss基本呈减小趋势，也就是模型在不断优化，在epoch为20的时候，模型基本已经收敛。在训练过程中，训练集的loss会比验证集上的loss小。



训练集上的loss曲线



验证集上的loss曲线

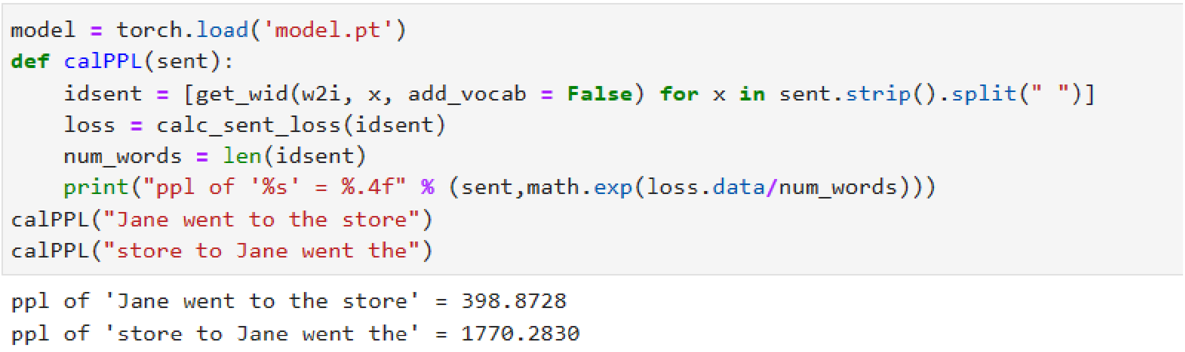
计算困惑度

对应代码如下：

```
model = torch.load('model.pt')
def calPPL(sent):
    idsent = [get_wid(w2i, x, add_vocab = False) for x in sent.strip().split(" ")]
    loss = calc_sent_loss(idsent)
    num_words = len(idsent)
    print("ppl of '%s' = %.4f" % (sent,math.exp(loss.data/num_words)))
calPPL("Jane went to the store")
calPPL("store to Jane went the")
```

首先将句子装换为字典序号，然后加载训练好的模型进行前向传播，计算交叉熵损失。最后，根据公式算出 ppl。

使用epoch为20的模型计算得到的困惑度值如下图所示：



```
model = torch.load('model.pt')
def calPPL(sent):
    idsent = [get_wid(w2i, x, add_vocab = False) for x in sent.strip().split(" ")]
    loss = calc_sent_loss(idsent)
    num_words = len(idsent)
    print("ppl of '%s' = %.4f" % (sent,math.exp(loss.data/num_words)))
calPPL("Jane went to the store")
calPPL("store to Jane went the")
```

```
ppl of 'Jane went to the store' = 398.8728
ppl of 'store to Jane went the' = 1770.2830
```