Q：梳理基于DNN-HMM的语音识别系统的流程，其有哪些步骤，每一步的输入、输出，步骤间的相互关系等，可以把自己的理解流程化、图形化、文字化的记录下来，写下来。

A：

最后一代的GMM-HMM将数据对齐打完标签后，递交给DNN网络进行有监督学习。Kaldi中dnn调用脚本：

local/nnet/run\_dnn.sh --stage 0 --nj $n exp/tri4b exp/tri4b\_ali exp/tri4b\_ali\_cv || exit 1;

现依次介绍该脚本中各步骤：

1. 生成fbanks特征（这个阶段生产DNN模型训练需要的fbank特征）

“steps/make\_fbank.sh”：输入“data/fbank/”下的原始数据，输出的fbank特征存放在“fbank/”文件夹下；

“steps/compute\_cmvn\_stats.sh“：对生成的fbank特征进行倒谱均值归一化处理；

1. 进行帧交叉墒训练（这个阶段时训练一DN来把帧分到对应的三音素状态）

“steps/nnet/train.sh”：这是通过mini-batch随机梯度下降法来做的。默认的是使用Sigmoid隐层单元，Softmax输出单元和全连接层AffineTransform。学习率是0.008，minibatch的大小是256；

“steps/nnet/decode.sh”：复用GMM生成的HCLG解码图对word和phone进行解码;

1. 训练dnn的序列辨别MPE/sMBR（这个阶段对所有的句子联合优化来训练神经网络，比帧层训练更接近一般的ASR目标）

“steps/nnet/align.sh”：将‘data’对齐到transition-ids序列上

“steps/nnet/make\_denlats.sh“：为MMI/MPE/sMBR 训练创建lattices(词图)，输入“$dir/lat.\*.ark,$dir/lat.scp“特征，输入词图。

1. 重复进行3轮MPE训练

“steps/nnet/train\_mpe.sh”：输入上面步骤中得到的对齐文件和词图，使用SGD梯度下降法，最大化从参考的对齐中得到的状态标签的期望正确率。

“steps/nnet/decode.sh”：复用GMM生成的HCLG解码图对word和phone进行解码;

约定->表示下一步，{}表示循环，[]表示括号内每一个都要进行一次，()表示不同分支下可能进行的操作，上述步骤绘制流程图如下：

{make\_fbank.sh->compute\_cmvn\_stats.sh}[train,dev,test]->train.sh->{decode.sh}[phone,word]->align.sh->make\_denlats.sh->train\_mpe.sh->{{decode.sh}[phone,word]}3