Automatic Speech Recognition 2017-18: Assignment

s1574336 s1619123

Task 1: Monophone models

In the following experiments for monophone models, we keep the default settings as shown in Table 1. We will clarify in each subsection if parameters in the settings will be changed.

FEATURES	CMVN	USE-ENERGY
MFCC(13) + D + D-D	ONLY CMN	FALSE

Table 1. The default settings for Monophone model experiments. D represents delta dynamic features.

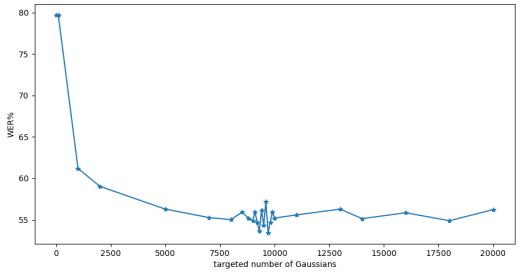

Figure 1. The variance of WER against the number of Gaussian mixture components.

1.1 Experiments With The Number of Gaussian

In order to investigate the influence of the number of Gaussians on the WER, we fixed other parameters but passed different number of Gaussians. At first, we changed the number exponentially, and observed that the model reached relatively low WER in the range between 9000 and 10000. Therefore, to optimize the number, we narrowed it down to this range and assessed the variance of WER at smaller intervals. The result is shown in Figure 1. It can be observed that with the increase in the number of Gaussian mixture components, the WER first decreases significantly, fluctuates in a certain range (9000 - 10000 in our case), then increases slightly after that. This result might have arisen because as the number of Gaussian mixture components increases, the degrees of freedom will also be increased and each component will model fewer data points. Hence, when there are not efficient data for each component, the model will assign too much probability on the training set and not enough probability on the unseen data in the test set. The optimal number of targeted Gaussians is 9700, and results at 9680 Gaussians after training.The corresponding WER is 53.42%.

Besides WER, we also investigated the log likelihood of training and test set. The average log likelihood of the training set is the output of the script, and for the log likelihood of test set we have taken the arithmetic average of all jobs. The results are shown in Figure 2 and Figure 3.

It can be observed that both the log likelihood of training and test sets increase with the number of Gaussians. After a certain point, the training log likelihood continues to increase while the test log likelihood levels off. This also confirms the assumption of overfitting according to the variance of WER. The execution time is generally linear with the number of Gaussians, but it can be observed that

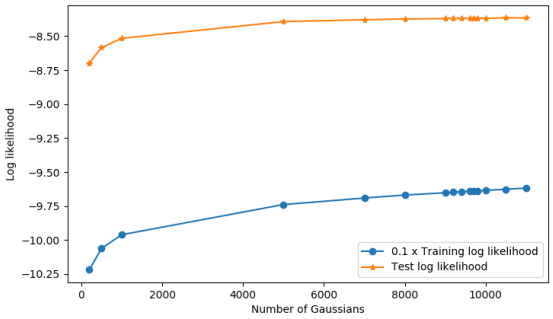

Figure 2. The variance of training and test log likelihood against the number of Gaussian mixture components. (As the training log likelihood is almost 10 times of test log likelihood, we scaled the training likelihood 10 times smaller to get better visualisation.)

when the number is small, there are several points that take an unexpectedly longer time. When the number is relatively small (less than 2000), the retrying probability in the alignment step is higher, around 0.3% - 0.6%, compared with 0.0% - 0.1% when the number is large. Therefore, apart from increasing the number of Gaussians, a small number of Gaussians leading to higher retry probability will also increase the execution time.

1.2 Experiments With Acoustic Features

In this question, we investigated the influence of two acoustic features—MFCC and PLP on WERs. We used the optimal number of Gaussians obtained from Task 1.1, which was 9700.

With regard to implementation and experiment settings, we

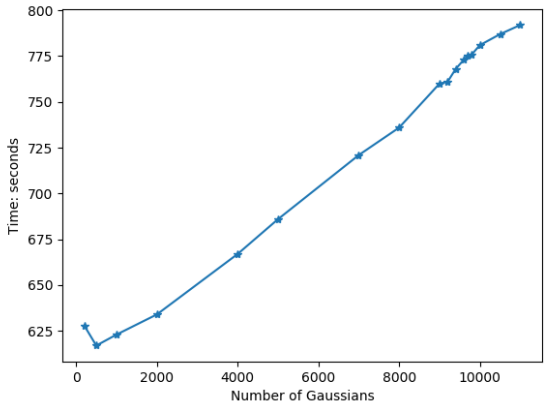

Figure 3. The variance of execution time along the number of Gaussian mixture components.

Acoustic Features	WER	train LL	test LL	Time
mfcc	53.42	-96.40	-8.37	701s
plp	54.44	22.16	0.91	694s

Table 2. Results (word error rate, log likelihood for both train and test sets, and time) of experiments using MFCC and PLP as acoustic features (with optimal number of Gaussians – 9700 and only CMN).

Acoustic Features	Cepstral Normalizations		
	no cmvn	cmn	cmvn
mfcc	58.40	**53.42**	**55.40**
plp	**55.91**	54.44	55.46

Table 3. The WERs of additional experiments with MFCC and PLP as acoustic features using different settings in cepstral mean/variance normalisation.

first extracted MFCC/PLP features of both training and test sets. The config files (mfcc.conf and plp.conf) for both features contain one line which sets user-energy to false. All other settings are consistent with the settings in the previous section. Next, we ran the mono model with 9700 number of Gaussians. We used the same categories for the results as before, the training log likelihood comes from the output of the script and the test log likelihood is calculated by averaging them over all jobs. The time measured is the total execution time including extracting each features and training the GMM model.

The initial result is shown in Table 2. As can be seen, the WER of PLP (54.44%) from our experiment is slightly higher than the value of MFCC (53.42%). However, as far as we know, theoretically PLP gives a slightly better result, especially under noisy conditions (Dave, 2013). Hence, we looked into a research study (LÃľvy et al., 2008) and we found that they conducted experiments of MFCC and PLP features with different settings such as various dimensions and CMVN settings. We then changed the CMVN settings for both features to investigate if different parameters change the results.

The results of extensional experiments are shown in Table 3. As can be seen, the WER varies when we change the conditions of CMVN settings. To be more specific, without CMVN, the PLP model yields a lower WER, whereas for other two conditions, the MFCC model generates lower WERs. Apart from the CMVN setting, it is also shown in another research paper (MÃijller & V. Psutka, 2001) that increasing the number of filters, from 9 to 17 for example, may lead to a worse performance of the PLP model compared to the MFCC model, which is consistent with our conclusion that the performance of using MFCC/PLP acoustic features varies with different settings.

1.3 Experiments With Delta Dynamic Features

In this section, we conducted experiments with different delta dynamic feature settings: MFCC, MFCC + delta, and MFCC + delta + delta-delta, to investigate how delta dynamic features of MFCCs affects WERs.

To implement the different dynamic feature settings, we modified two scripts—train_mono.sh and decode.sh, by adding an additional argument for delta_opts. We set delta-order from 0 to 2, where 0 means no delta dynamic features are added for MFCC, 1 is MFCC + delta, and 2 is MFCC + delta + delta-delta. In terms of other parameters, we adopted the optimal number of Gaussians from Task 1.1 and used the default setting for CMVN – only the mean gets normalised.

Here is the results from our three experiments (see table 4). It can be observed that the WER is significantly higher without using delta as dynamic features (81.34%). Also, compared to computing only one delta, the result obtained from delta-delta achieved the lowest WER (53.42%).

In terms of delta features, we know that adding dynamic information to the static cepstral features improves the performance, since the spectral dynamics are emphasised by the linear combination of the polynomial expansion coefficients in delta feature computation (Furui, 1986). This is also reflected in our results that adding delta feature lowers the WER by at least 20% in our experiments.

With regard to using delta and delta-delta, our results show that the latter yields a lower WER (–7.09%) compared to the former. One potential explanation could be that delta MFCC causes problems when the environment is highly reverberant and there is very little ambient noise (Ichikawa et al., 2010). Another research study (Kumar et al., 2011) agrees with this by showing empirically that delta-delta outperforms delta with an increase in Signal-to-Noise Ratio

Dynamic Feature Settings	WER	Train log likelihood	Test log likelihood
mfcc	81.34	-47.42	-4.18
mfcc + delta	60.51	-77.91	-6.79
mfcc + delta + delta-delta	53.42	-96.40	-8.37

Table 4. Results (word error rate, log likelihood for both training and test sets) of experiments with delta as dynamic features (with optimal number of Gaussians – 9700).

(SNR) ranging from 0 to 20.

1.4 Experiments With CMN/CVN

In this section, we conducted experiments with different settings of cepstral mean and variance normalisation (no normalisation, only CMN, both CMN and CVN), to investigate how cepstral mean and variance normalisation of MFCC influence WER. We omitted the situation which only normalised the variance without normalising the mean. The reasons for that are: firstly, the standard procedure for variance standardisation is based on centring the data, secondly, kaldi does not support normalising only the variance but not the mean.

To implement different mean/variance normalisation settings, we modified train_mono.sh and decode.sh by adding two additional arguments for norm_means and norm_vars, which then modifies the settings for cmvn_opts. In terms of other parameters, we adopted the optimal number of Gaussians from Task 1.1. The corresponding results are shown in the following Table 5, where the model with only mean normalisation obtains the best WER on the test set.

Cmn / Cvn	WER	train/test log likelihood
no cmvn	58.40	-96.65 /-8.40
only cmn	53.42	-96.40 /-8.37
cmn + cvn	55.40	6.50 /0.21

Table 5. Results (word error rate, log likelihood for both training and test sets) of experiments with mean and variance normalisation (with MFCC + delta + delta-delta features, optimal number of Gaussians – 9700)

It can be observed that using the original data gives the worst performance. Contrary to expectations, normalising both of mean and variance doesn't give the best result in terms of WER. That might be because the noise of the original distribution was unexpectedly enlarged when scaling the data. It has been published that standardisation is always an open issue that we can use different standardisations when dealing with different data sets (Greenspan et al., 2006; Craig et al., 2006). Therefore, we can conclude that for this specific dataset in this assignment, only normalising cepstral mean leads to the best performance.

However, normalising both the mean and variance gives the highest log likelihood. It suggests although we maximise log likelihood when training, it is not efficient to use this as the only metric to evaluate the model.

Task 2: Tied-state Triphone Models

In this section, we carry out experiments to investigate how the number of clusters and Gaussian mixture components influence the WER.

To implement different settings for number of clusters and Gaussians, we passed different numbers to train_delta.sh. The relevant arguments in the script are *numleaves* and *totgauss*, where *numleaves* represents the targeted maximum clusters, and *totgauss*/*numleaves* represents the targeted average number of Gaussian components for each Gaussian mixture model. Therefore, by fixing *totgauss*/*numleaves* to different numbers, we investigated how the number of clusters influence the WER with the same targeted average Gaussian components, and vice versa. We firstly tried setting *numleaves* ranging from 1000 to 7000, and *totgauss*/*numleaves* ranging from 2 to 20. Then we noticed that the performance does not improve when the *numleaves* is larger than 4000 or *totgauss*/*numleaves* is larger than 15. Therefore, we narrowed down to that region and the experimental results are shown in Table 6. The local optimal configuration of parameters conditional on our experiments are: *numleaves* is 1000 and *totgauss*/*numleaves* is 10 (or *totgauss* is 10000)

It can be observed that when there are fewer clusters, which means we are trying to cluster more states into one group, a small number of Gaussian components performs poorly, as it might underfit the data with limited degrees of freedom. Therefore, when we increase the number of Gaussian components, WER will decrease initially but then increase again due to the overfiting problem discussed in Task 1.1. In addition, it can be also noticed, the model will encounter an overfitting problem sooner with more clusters. For example, WER for the model with a targeted number of 3000 clusters stops decreasing when there are 6 Gaussian components, while that for the model with 1000 clusters is still improving up to 10 Gaussian components.

		Average Gaussian Components per GMM						
		2	3	6	9	10	12	15
clusters	1000	50.29	48.37	48.56	45.37	**45.05**	45.62	45.43
	2000	49.01	46.96	46.07	46.65	45.62	46.52	47.99
	3000	48.12	46.45	46.65	47.54	47.03	47.48	48.69
	4000	48.05	46.96	46.71	50.16	48.69	48.39	48.58

Table 6. WER results of experiments with different number of clusters and Gaussian components(with MFCC + delta + delta-delta features).

Feature Fransformations	Train file	Decode file	WER	Train/Test log likelihood
baseline	train_deltas	decode	45.05	-96.29 /-8.38
lda mllt	train_lda_mllt	decode	**43.77**	-49.33 /-4.49
lvtln	train_lvtln	decode_lvtln	44.28	-95.23 /-8.41

Table 7. Settings (train and decode files used) and results of experiments with different feature transformations(with MFCC + delta + delta-delta features) . The baseline is the best triphone model that we obtained from Task 2.

Task 3: Advanced tasks

In this section, we investigate two different areas—**Feature Transformations** and **Neural Networks** for our advanced tasks. Our goal is to explore to what extent these two techniques improve WER. The script for feature transformations is exp_adv_t1.sh and the script for neural networks is exp_adv_t2.sh. In terms of baseline, we used the triphone model that we built in Task 2, which has a WER of 45.05%..

3.1 Feature Transformations

In this subsection, we investigate the how the feature transformations influence WER. Since Kaldi supports a number of feature and model-space transforms, we selected **Linear Discriminant Analysis (LDA) and Maximum Likelihood Linear Transform (MLLT)** from the non-speaker specific transformation group and **Linear Vocal Tract Length Normalization (LVTLN)** transformation from global transform category which is typically applied in a speaker adaptive way.

We started from the baseline model that we built in Task 2, where no feature transformations are involved. For a better comparison, we set the optimal number of leaves and the number of Gaussians to 1000 and 10000 respectively, which were obtained from Task 2. The dynamic features (delta + delta-delta) and acoustic feature (MFCC) are also set to the same as in Task 2.

Regarding to implementations, it appears that in Kaldi we need different training and decoding files for building our models. We then explored under the Kaldi directory *steps*, and found that for LDA + MLLT transformations, we can use files train_lda_mllt.sh for the training process and decode.sh for the decoding process. Similarly, for LVTLN transformation, we can use train_lvtln for training and decode_lvtln for decoding.

The results we obtained are shown in Table 7. As can be seen, with feature transformations, the WER decreased by 1.28% for LDA + MLLT transformations and 0.77% for the LVTLN transformation. Our results suggest that feature transformations improve WER of our triphone models.

Psutka (V., 2007) listed the benefits of using MLLT combined with widely used transforms (LDA etc.), amongst which included a drastic reduction of WER, and this is consistent with our results. Moreover, the core idea behinds the feature transformations is the reduction in dimensionality of the features. Thus, LDA + MLLT can also accelerate the training process. As for the Linear VTLN transformation, a number of studies (Kim et al., 2004; Rath et al., 2009) have also shown that this linear transformation improves the performance due to its high efficiency in terms of warp factor estimation and application of the warp factors.

3.2 Neural Networks

In this subsection, we investigate the how neural networks influence WER compared to non-neural network structures. Furthermore, we also explore the impacts on WER from different neural network architectures.

We used nnet1 (http://kaldi-asr.org/doc/dnn1.html) which is a deep neural network structure provided by Kaldi. We used the default division for the training (90%) and validation sets (10%) with the subset_data_dir_tr_cv.sh script. Next, we trained the neural network with the steps/nnet/train.sh script based on the alignment generated by the best model in Task 2 (1000 *numleaves*, 10000 *totgauss*). Other experiments settings are shown in Table 9. The results can be seen from Table 8: the lowest WER (43.00%) is achieved by the model that contains 2 hidden layers, 256 hidden dimensions, which is 3.05% lower compared to WER of our baseline. The results suggests an improvement when using neural networks. A few recent studies (Graves et al., 2013;

LAYERS	HIDDEN DIM	WER	TRAIN/VAL LOSS (XENT)	TEST LOG LIKELIHOOD	TIME(SECONDS)	ITERATIONS
2	128	45.37	2.25 / 2.45	1.83	5453	14
2	256	**43.00**	1.97 / 2.33	1.92	6269	13
2	512	43.07	1.75 / 2.26	1.95	9081	13
1	256	44.92	2.04 / 2.39	1.89	6281	13
3	256	43.19	1.97 / 2.32	1.11	6878	13

Table 8. Results of experiments with different neural network structures (with MFCC + delta + delta-delta features).

LAYERS	HIDDEN DIM	LEARNING RATE	NN TYPE	SPLICES
1, 2, 3	128, 256, 512	0.008	DNN	5

Table 9. Experiments setting for neural network structures in Task 3.2.

Hinton et al., 2012) also argue that neural networks lead to significant advances in automatic speech recognition, which is consistent with our results.

It can be noticed that increasing the number of layers or hidden dimensions can both improve the WER. It makes sense as the growth of parameters makes the model more flexible to fit the data. We also observed that the model stops training when there is no obvious improvements, this early stopping mechanism provided in Kaldi can be used to prevent overfitting. However, when we increased the number of layers to 3, it does not improve the WER performance compared with the best model, but still has lower training loss. The reason for that might be the the improvement at that iteration stage is small and early stopping prevents it from training the model further. It might also result from the difference between the metrics (WER and loss), as the neural network is minimising the loss instead of WER.

On the other hand, also observed from the results, the training time taken is significantly higher in comparison with non-neural network structures. Moreover, an increase in the number of layers and hidden dimensions also add on to the processing time. Hence, we conclude that the main drawback of using neural networks is that the process is time-consuming and the computation cost is proportional to the growth of the parameters.

References

Craig, Andrew, Cloarec, Olivier, Holmes, Elaine, Nicholson, Jeremy K., and Lindon, John C. Scaling and normalization effects in nmr spectroscopic metabonomic data sets. *Analytical Chemistry*, 78(7):2262–2267, 2006. doi: 10.1021/ac0519312. URL https://doi.org/10.1021/ac0519312. PMID: 16579606.

Dave, Namrata. Feature extraction methods lpc, plp and mfcc in speech recognition. Volume 1, 07 2013.

Furui, S. Speaker-independent isolated word recognition based on emphasized spectral dynamics. In *ICASSP '86. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 11, pp. 1991–1994, Apr 1986. doi: 10.1109/ICASSP.1986.1168654.

Graves, A., r. Mohamed, A., and Hinton, G. Speech recognition with deep recurrent neural networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 6645–6649, May 2013. doi: 10.1109/ICASSP.2013.6638947.

Greenspan, H., Ruf, A., and Goldberger, J. Constrained gaussian mixture model framework for automatic segmentation of mr brain images. *IEEE Transactions on Medical Imaging*, 25(9):1233–1245, Sept 2006. ISSN 0278-0062. doi: 10.1109/TMI.2006.880668.

Hinton, G., Deng, L., Yu, D., Dahl, G. E., r. Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., and Kingsbury, B. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, Nov 2012. ISSN 1053-5888. doi: 10.1109/MSP.2012.2205597.

Ichikawa, O., Fukuda, T., and Nishimura, M. Dynamic features in the linear-logarithmic hybrid domain for automatic speech recognition in a reverberant environment. *IEEE Journal of Selected Topics in Signal Processing*, 4(5):816–823, Oct 2010. ISSN 1932-4553. doi: 10.1109/JSTSP.2010.2057191.

Kim, Do Yeong, Umesh, S., Gales, Mark J. F., Hain, Thomas, and Woodland, Philip C. Using vtln for broadcast news transcription. In *INTERSPEECH*, 2004.

Kumar, K., Kim, C., and Stern, R. M. Delta-spectral cepstral coefficients for robust speech recognition. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4784–4787, May 2011. doi: 10.1109/ICASSP.2011.5947425.

LÃl'vy, Christophe, LinarÃís, Georges, and Nocera, Pascal. Comparison of several acoustic modeling techniques and decoding algorithms for embedded speech recognition systems. 12 2008.

MÃijller, LudÄŻk and V. Psutka, Josef. Comparison of mfcc and plp parameterizations in the speaker independent continuous speech recognition task. pp. 1813–1816, 01 2001.

Rath, Shakti, Umesh, Srinivasan, and Sarkar, Achintya. Using vtln matrices for rapid and computationally-efficient speaker adaptation with robustness to first-pass transcription errors. pp. 572–575, 01 2009.

V., Psutka Josef. Benefit of maximum likelihood linear transform (mllt) used at different levels of covariance matrices clustering in asr systems. *Lecture Notes in Artificial Intelligence*, pp. 431–438, 2007. URL http://www.kky.zcu.cz/en/publications/PsutkaJosefV_2007_Benefitofmaximum.