## Experiments report

**PS：红色表示需要修改的地方**

**最新更新在最后几页**

### Introduction

The goal of this work is to improve SRE performance in PLDA score using Variational Autoencoder and triplet loss.

Probabilistic linear discriminant analysis (PLDA) is the defector standard for backends in i-vector speaker recognition. But if we try to extend the PLDA in d-vector or x-vector SRE system, the performance is not as well as i-vector. One of the main causes of this problem is that i-vector follows Gaussian distribution, but d-vector and x-vector do not.

Some predecessors (Lilt...) think that there are 2 main factors determine the PLDA performance:

**1. the Gauss property of embedding**

**2. Distinguishability between different speaker's embedding**

In this work, we propose to approach this problem using stochastic gradient variational Bayes, which aims to enhance the Gauss property of d/x-vector. And we also add triplet loss into model loss function to increase the distinguishability between different speaker's embedding.

用中文重新表述一下·

决定vector在PLDA的好坏有两个因素，

1. vector是否有很强的高斯性
2. 不同说话人之间的vector是否有很强的区分性

这个实验的目标：

用VAE使vector高斯+降维（但可能会降低区分性）

因此用triplet使得vector保证有区分性

总结来说，用VAE使vector变得高斯的情况下用triplet loss保证区分性，从而提高PLDA的性能

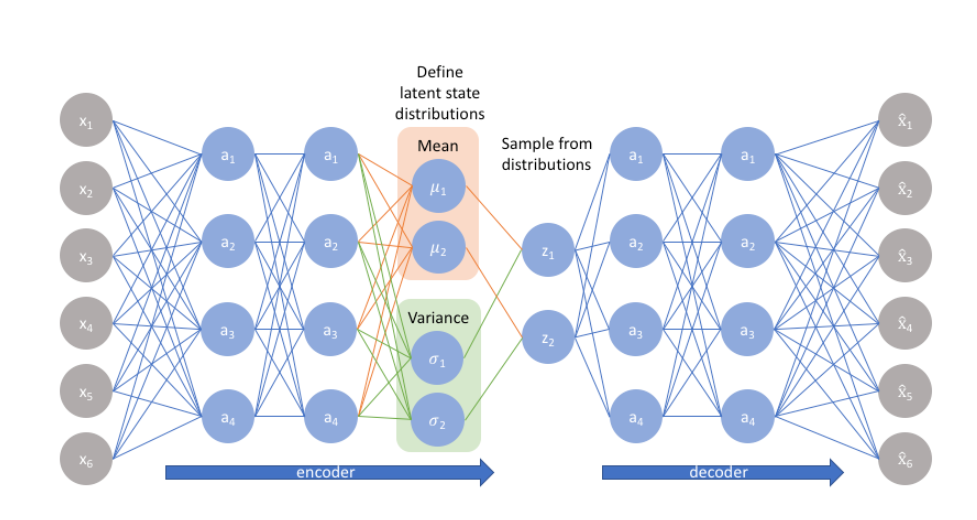
### Experiments

#### Network structure

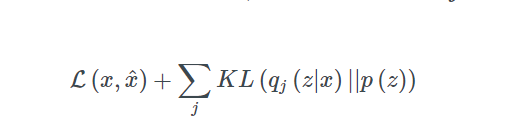
###### Vanilla VAE

VAE encoder: 2 hidden layers (sigmoid+tanh)

VAE decoder: 2 hidden layers (tanh+sigmoid)

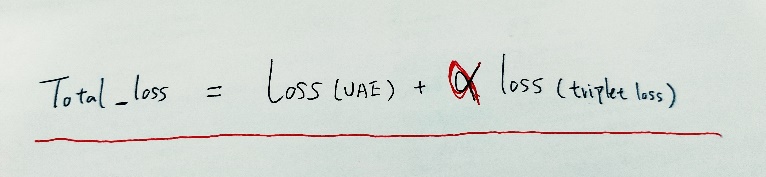


We trained this model using **reparameterization trick**, this enables us to optimize of the loss function using backpropagation to jointly estimate the parameters that define the model and the approximate posteriors of the latent factors



###### VAE with Triplet Loss

结构图略



#### Work Implementation

1. Firstly, we use kaldi toolkit to extract d-vector and x-vector with 512 dimensions from utterance.
2. Then, we implemented the VAE with the TensorFlow toolkit, the inputs are d-vector and x-vector extracted from the first step.
3. When we finish the train training process, we input the original d/x-vector into network and get the output from VAE encoder.
4. At last, we use kaldi toolkit to calculate EER of output vector.

NOTE: I have submitted the TF-VAE code to GitHub. (<https://github.com/zyzisyz/zkhhk>)

#### Datasets

**Train**: voxceleb\_combined\_200000

**Test**: sitw\_dev & sitw\_eval

#### Baseline

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Skew | Kurt | Cosine EER | PLDA EER | LDA PLDA |
| d-vector | 0.1092 | -0.11609 | 38.39% | 17.71% | 9.511% |
| x-vector | -0.04228 | -0.3603 | 15.67% | 9.087% | 3.157% |
| d-vector-LDA | -0.005515 | 0.028395 | 12.94% | 9.665% | 9.434% |
| x-vector-LDA | -0.01074 | -0.0089 | 5.198% | 3.735% | 3.157% |

#### Experiment Results:

We designed 4 groups of experiments to find the best network parameters.

1. D/X-vector -> Vanilla VAE
2. D/X-vector -> VAE with Triplet Loss
3. D/X-vector -> LDA -> Vanilla VAE
4. D/X-vector -> LDA ->VAE with Triplet Loss

以下实验结果为KL和MSE等权重

###### D/X-vector -> Vanilla VAE

Vanilla VAE Best Result

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | The Best Network structure | Cosine EER | | PLDA EER | | LDA PLDA EER | |
| baseline | VAE | baseline | VAE | baseline | VAE |
| d-vector | Z dim: 600  layer unit: 1800 | 38.39% | **38.54%** | 17.71% | **16.02%** | 9.511% | **12.59%**  **(75)** |
| x-vector | Z dim: 200  layer unit: 2300 | 15.67% | **13.79%** | 9.087% | **5.006%** | 3.157% | **4.236%**  **(100）** |

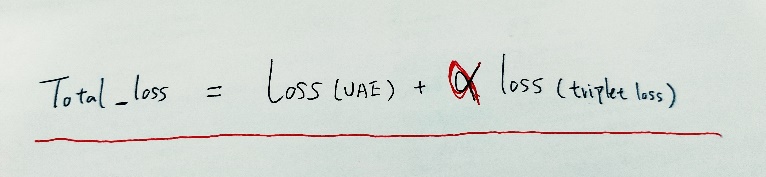
The PLDA EER of d-vector does not have an obvious enhancement. But the X-vector’s result seems good.

Note:

1. Baseline means without VAE encoder.
2. The best d-vector-VAE’s Z dim is 600 which is more than 512.
3. LDA PLDA EER 括号表示LDA的dim

###### D/X-vector -> VAE with Triplet Loss

The total loss function of VAE with Triplet Loss consists of 2 parts ( VAE Loss + Triplet Loss), in this experiment, we use the Vanilla VAE best network structure we have got before to find the best hyperparameter α.



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | The Best Network structure | Cosine EER | | PLDA EER | | LDA PLDA EER | |
| Baseline | VAE | baseline | VAE | baseline | VAE |
| d-vector | α: 150 | 38.39% | **16.1%** | 17.71% | **13.67%** | 9.511% | **11.17%**  **(65)** |
| x-vector | α: 200 | 15.67% | **6.546 %** | 9.087% | **4.197%** | 3.157% | **4.043%**  **(130)** |

###### D/X-vector -> LDA -> Vanilla VAE

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | The Best Network structure | Cosine EER | | PLDA EER | | LDA PLDA EER | |
| baseline | VAE | baseline | VAE | baseline | VAE |
| d-vector | Z dim: 80  layer unit: 1600 | 12.94% | **12.09%** | 9.665% | **9.472%** | 9.434% | **9.049%（70）** |
| x-vector | Z dim: 120  layer unit: 1800 | 5.198% | **4.39%** | 3.735% | **3.581%** | 3.157% | **3.273%**  **（110）** |

###### D/X-vector -> LDA ->VAE with Triplet Loss

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | The Best Network structure | Cosine EER | | PLDA EER | | LDA PLDA EER | |
| baseline | VAE | baseline | VAE | baseline | VAE |
| d-vector | α=30 | 12.94% | **10.74%** | 9.665% | **9.472%** | 9.434% | **9.203%**  **(85)** |
| x-vector | α=6 | 5.198% | **4.736%** | 3.735% | **3.926%** | 3.157% | **3.735%**  **（80）** |

最终，我和蓝天学长选择的网络结构为

**512(输入)->1800->1800->200VAE(输出)**

**512(输入)->LDA->150-> 1600->1600->100(输出)**

**总的来说，以上实验结果表明，**

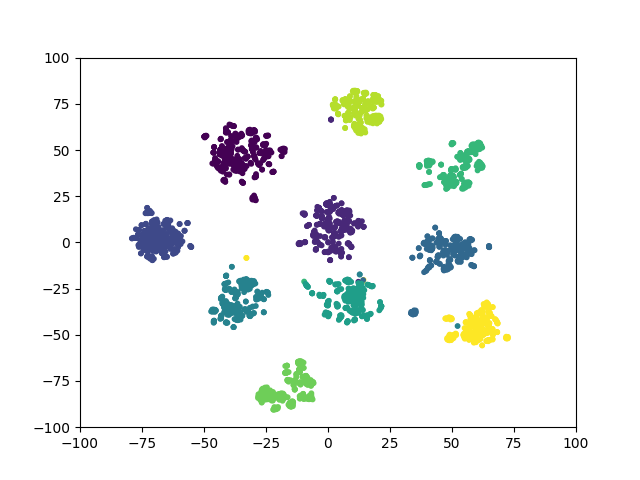
1. **VAE在x-vector的PLDA EER上有着非常明显的作用**
2. **Triplet Loss 在Cosine EER有着非常明显的作用**
3. **vector本身的区分性已经很强了，Triplet Loss的Loss值很小，小于1，因此所乘的权重很高，我设置的为100。**

**X-vector的Tsne图**

**X**

Skew: -0.04228, Kurt: -0.3603

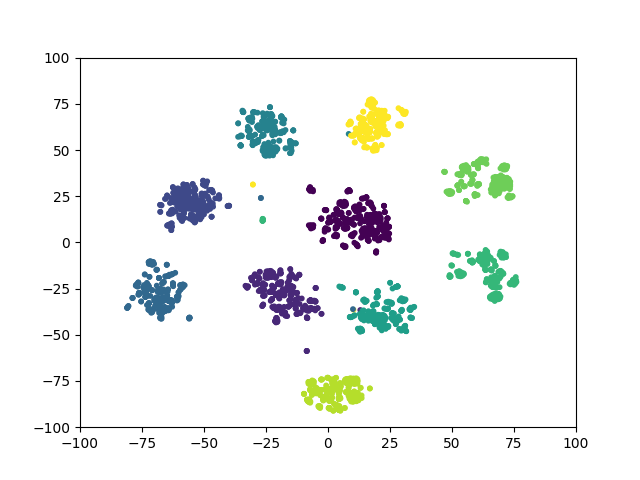
Cosine EER: 15.67%: , PLDA EER: 9.87%



**X-VAE**

Skew: -0.00436, Kurt: 0.0346

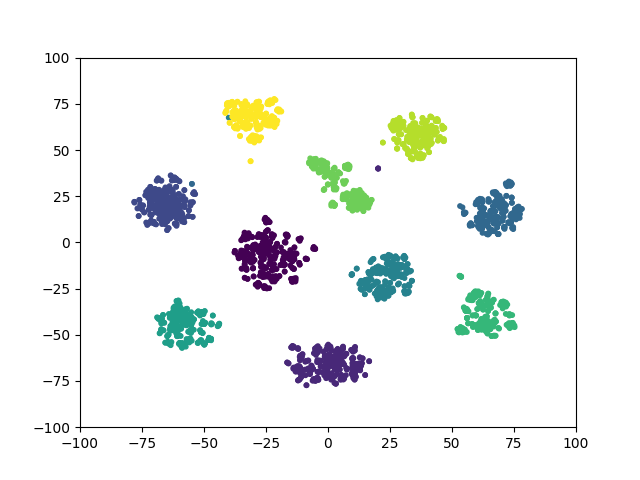
Cosine EER: 13.76%, PLDA EER: 5.006%



**X-VAE\_TL**

Skew: -0.01047, Kurt: -0.03296

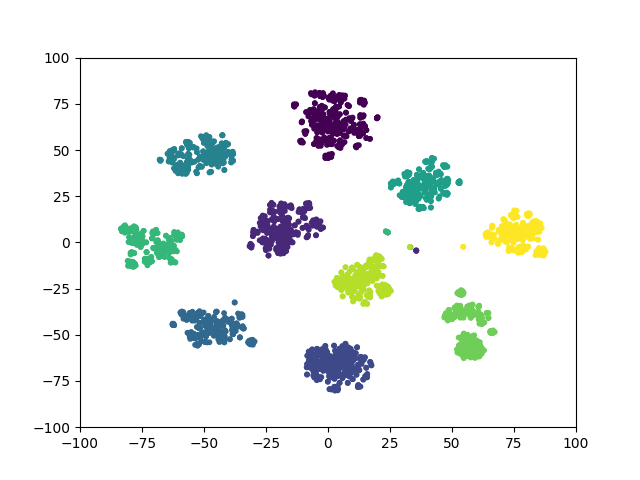
Cosine EER: 6.252% PLDA EER: 4.179%



**X-LDA**

Skew: -0.0107, Kurt: -0.00894

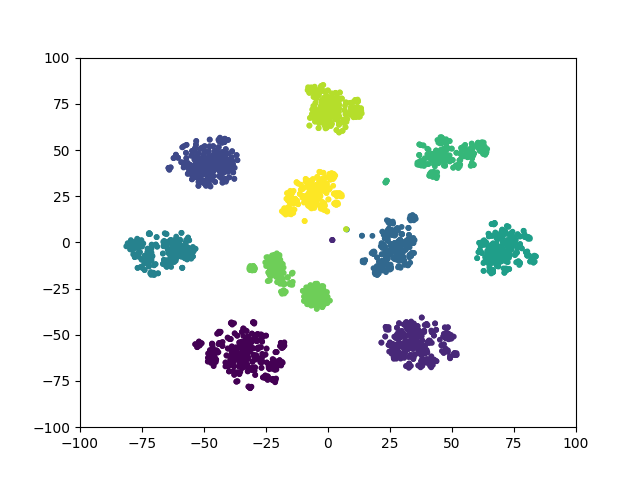
Cosine EER: 5.198% PLDA EER: 3.735%



**X-LDA-VAE**

skew: 0.00277, Kurt: -0.0378

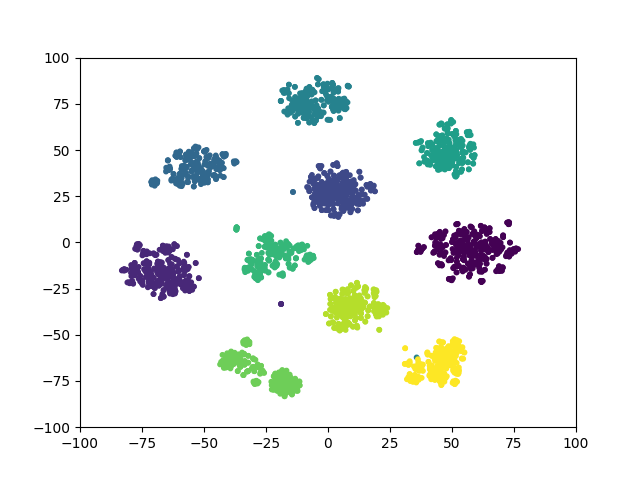
Cosine EER: 4.39% , PLDA EER: 3.581%



**X-LDA-VAE\_TL**

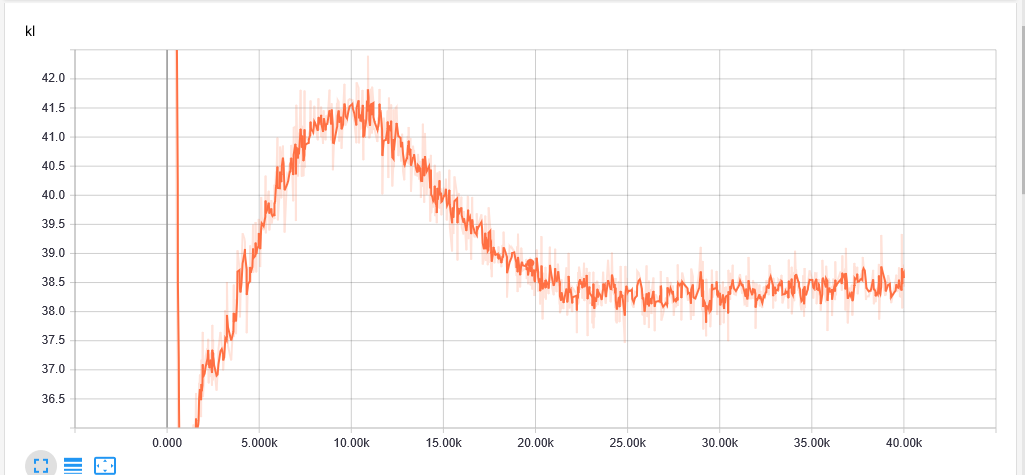
Skew: 0.00291, Kurt: -0.0536

Cosine EER: 4.736%, PLDA EER: 3.926%



但实验中我们发现，

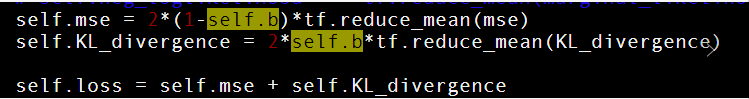
1. KL loss收敛的很奇怪



1. 因此修改了Vanilla VAE中KL所占的权重，实验发现，当KL权重很低时，即MSE所占权重很高时，PLDA EER性能更好。
2. Triplet loss可以把Cosine EER 降得很低，但PLDA EER反而没有纯VAE的好

以下实验结果为

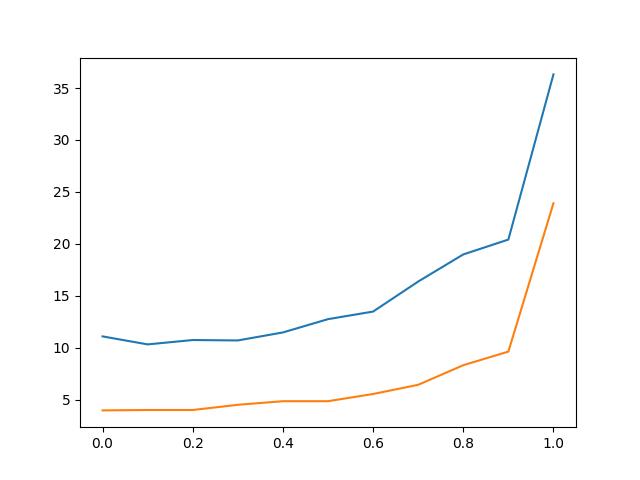
**寻找KL和MSE最优权重所做实验**



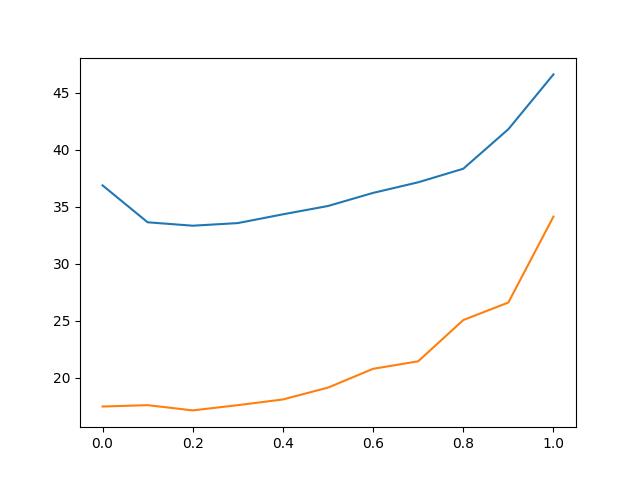
**横坐标是KL所占权重，纵坐标是PLDA EER 和Cosine EER**

**当横坐标是0.5，KL和MSE是等权重**

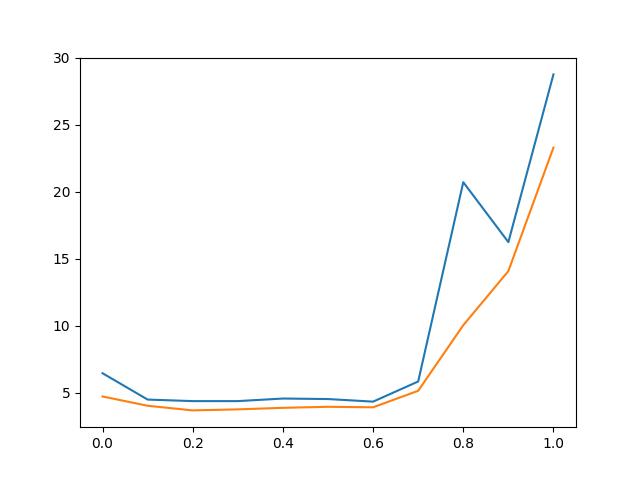
**X-vector 训练了40个epoch 每个epoch 1000个batch**



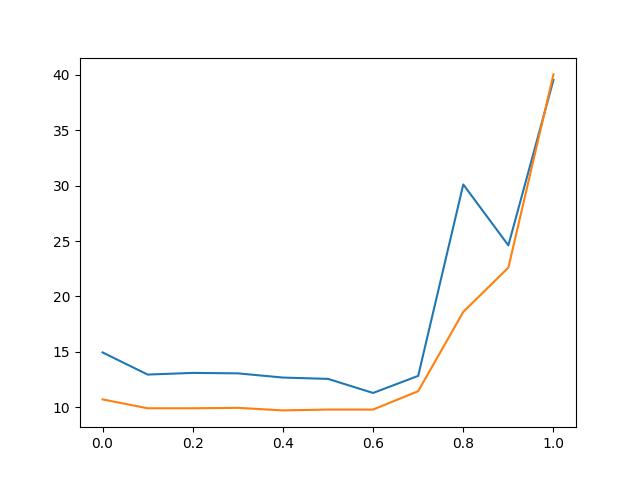
**D-vector 训练了40个epoch 每个epoch 1000个batch**



**X-vector经过LDA后训练了40个epoch 每个epoch 1000个batch**



**D-vector经过LDA后训练了40个epoch 每个epoch 1000个batch**



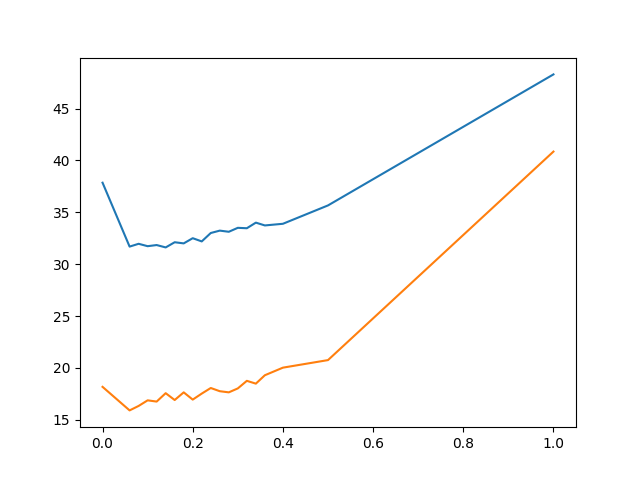
**从上面四张图，我们发现KL的权重应该小于0.5**

**因此我在0.06与0.5之间做了更加精细的实验**

**注：实验结果显示KL散度有用，但权重不能太高，得非常低**

**感觉Loss还没完全收敛，我还需要再测试一下**

**但大致的图为**



**结果**

1. **VAE PLDA EER上比Baseline好很多，但最好的还是直接LDA PLDA**
2. **VAE PLDA 比 baseline的PCA PLDA 好**
3. **VAE有一点点或者可以说没有 降维功能**
4. **PCA PLDA的效果没有LDA PLDA的好，有些时候甚至比直接PLDA的结果还差**
5. **加上Triplet Loss训练VAE，可以非常大的降低Cosine EER（甚至比PLDA EER还低）**
6. **KL散度同triplet loss 和 MSE 中任意一支or多支loss配合，都可以比baseline的纯PLDA结果好。但只有KL效果会很差**
7. **限制sigma=0.1比较好，但没有纯VAE的结果好。（其他取值的sigma还在训练寻找）**

X-vector

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Cosine | PLDA | PCA PLDA | LDA PLDA |
| Basic X-vector | 15.67% | 9.087% | 4.197% | 3.157% |
| LDA | 5.198% | 3.735% | 3.312% | 3.157% |
| VAE | 9.357% | 3.774% | 4.813% | 3.389% |
| LDA VAE | 4.39% | 3.581% | 3.389% | 3.273% |
| DAE | 17.87% | 4.736% | 5.083% | 3.581% |
| VAE Triplet Loss | 5.776% | 3.735% | 4.21% | 3.535% |
| Auto Encoder | 20.52% | 17.6% | 4.967% | 4.197% |
| MLP Triplet Loss | 5.391% | 7.201% | 4.852% | 4.659% |

D-vector

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Cosine | PLDA | PCA PLDA | LDA PLDA |
| Basic D-vector | 38.39% | 17.71% | 16.21% | 9.511% |
| LDA | 12.94% | 9.511% | 9.28% | 9.087% |
| VAE | 28.3% | 14.32% | 14.48**%** | 10.67% |
| LDA VAE | 14.29% | 10.55% | 9.973% | 9.626% |
| DAE | 37.31% | 19.21% | 16.4% | 10.74% |
| VAE Triplet Loss | 13.28% | 13.4% | 11.7% | 10.05% |
| Auto Encoder | 39.28% | 23.99% | 17.25% | 14.32% |
| MLP Triplet Loss | 12.63% | 15.75% | 11.59% | 11.59% |

VAE X-vector dim

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Cosine | PLDA | PCA PLDA (100) | LDA PLDA (100) |
| Basic X-vector | 15.67% | 9.087% | 4.197% | 3.157% |
| Z: 200 | 9.357% | 3.774% | 4.813% | 3.504% |
| Z: 512 | 9.395% | 4.005% | 4.89% | 3.427% |

512的比200的差一点，但差距不是很明显

VAE D-vector dim

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Cosine | PLDA | PCA PLDA (100) | LDA PLDA (100) |
| Basic D-vector | 38.39% | 17.71% | 16.21% | 9.511% |
| Z: 200 | 26.84% | 14.05% | 15.06% | 10.51% |
| Z: 512 | 26.26% | 15.02% | 14.75% | 10.47% |

0.1 的时候结果最好

X-Vector Unified sigma

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Cosine | PLDA | PCA PLDA (100) | LDA PLDA (100) |
| Baseline | 15.67% | 9.087% | 4.197% | 3.157% |
| VAE | 9.357% | 3.774% | 4.813% | 3.389% |
| Sigma: 1 | 11.51% | 5.006% | 4.082% | 3.735% |
| Sigma: 0.1 | 11.28% | 3.812% | 4.197% | 3.427% |
| Sigma: 0.01 | 12.01% | 3.851% | 4.351% | 3.62% |

0.1 的时候结果最好

D-Vector Unified sigma

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Cosine | PLDA | PCA PLDA | LDA PLDA |
| Baseline | 38.39% | 17.71% | 16.21% | 9.511% |
| VAE | 28.3% | 14.32% | 14.48**%** | 10.67% |
| Sigma: 1 | 33.35% | 17.02% | 15.86% | 13.21% |
| Sigma: 0.1 | 32.65% | 14.86% | 16.79% | 10.78% |
| Sigma: 0.01 | 33.54% | 15.33% | 18.37% | 11.4% |