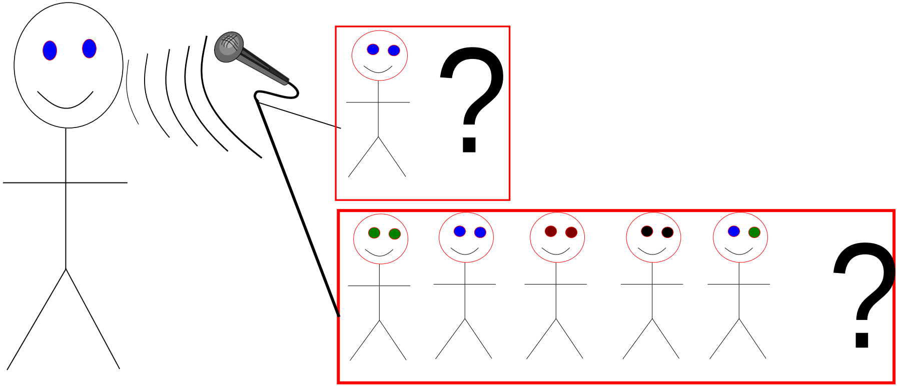
DeepVoice

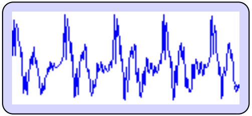
Extracting meaningful signal representation for Speaker Recognition using deep architectures

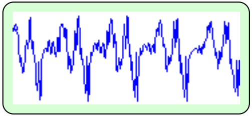
Rémi Hutin, Rémy Sun, Raphaël Truffet
Supervisors : Guillaume Gravier and Vedran Vukotić

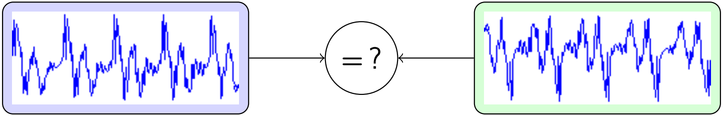
Computer science department
ENS Rennes

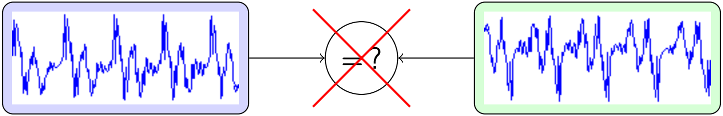
Linkmedia project
IRISA

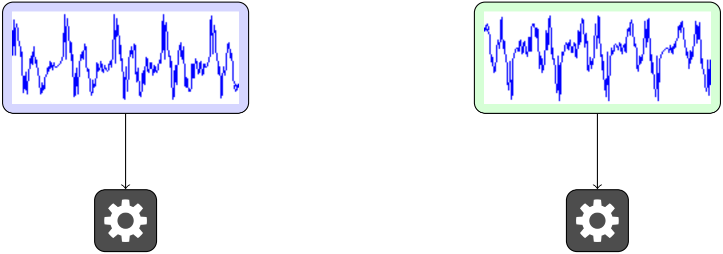


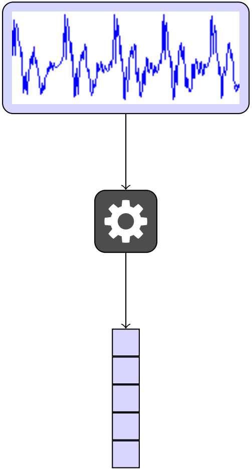


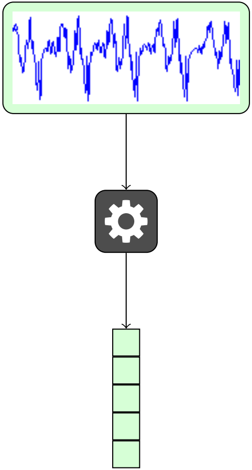


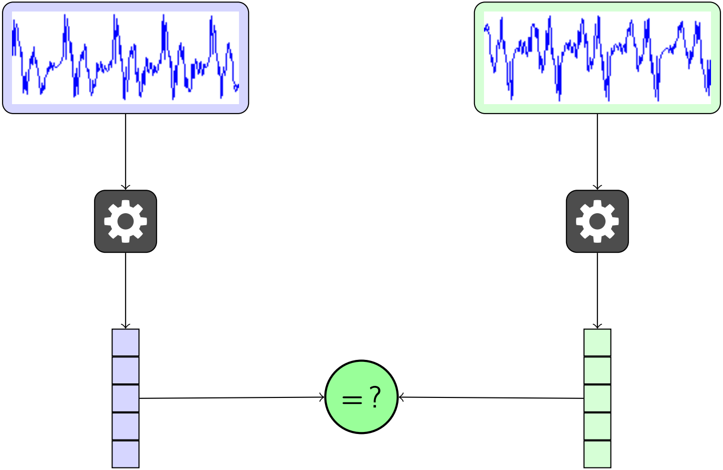


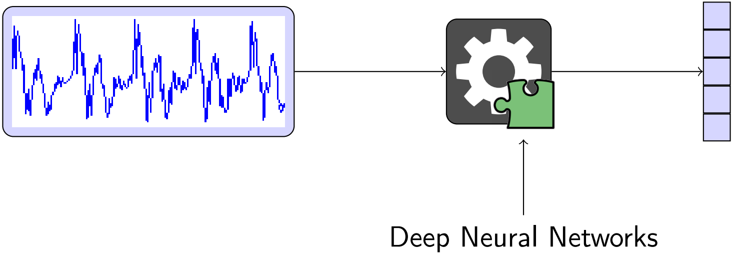








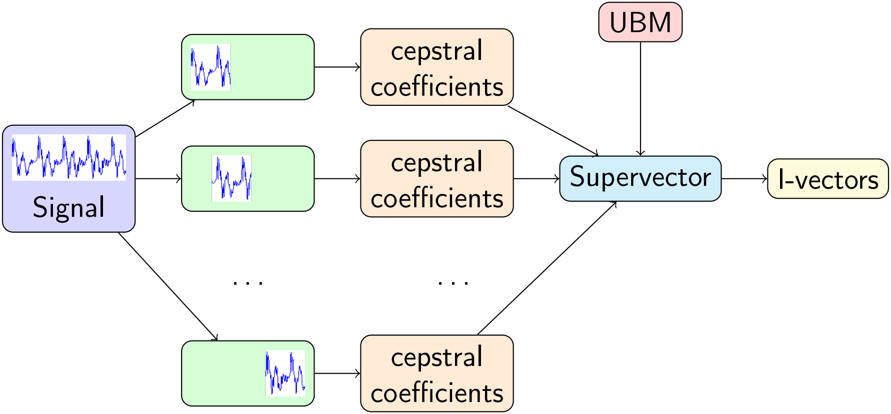



Deep Neural Networks

Outline

Outline

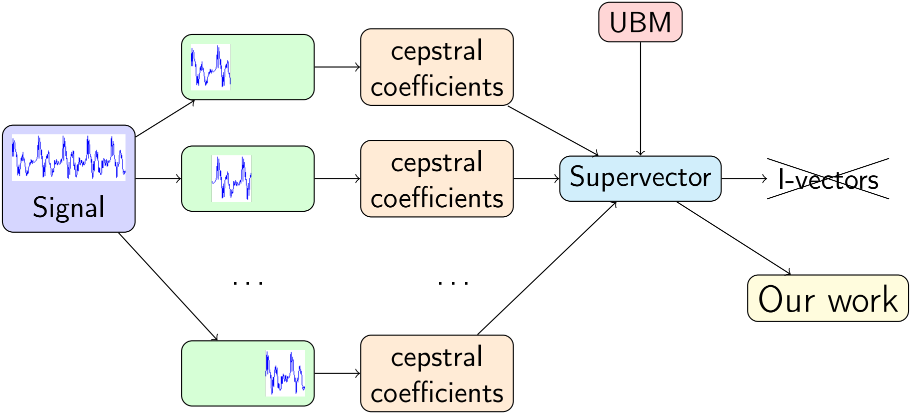
Signal processing workflow



Question

Can we do better than i-vectors ?

Signal processing workflow



Outline

Use of Deep Neural Networks (DNN)

Deep neural networks are interesting because :

- **Non-linear** feature extraction

Use of Deep Neural Networks (DNN)

Deep neural networks are interesting because :

- **Non-linear** feature extraction
- They naturally generate several level of representation

Use of Deep Neural Networks (DNN)

Deep neural networks are interesting because :

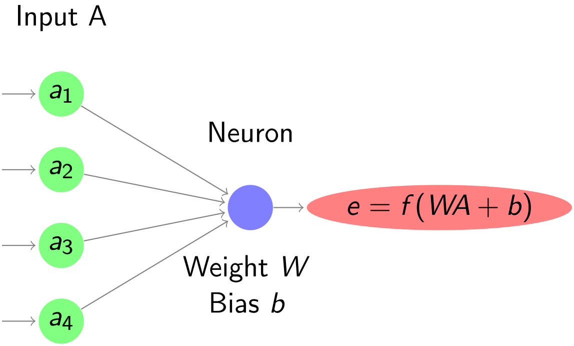
- **Non-linear** feature extraction
- They naturally generate several level of representation
- They bring out unsuspected features

Use of Deep Neural Networks (DNN)

Deep neural networks are interesting because :

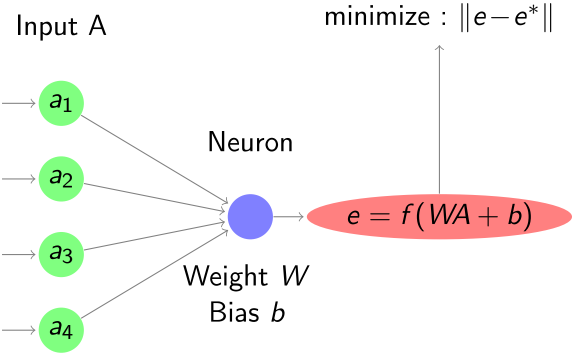
- **Non-linear** feature extraction
- They naturally generate several level of representation
- They bring out unsuspected features
- There is a multitude of architectures

Formal neuron



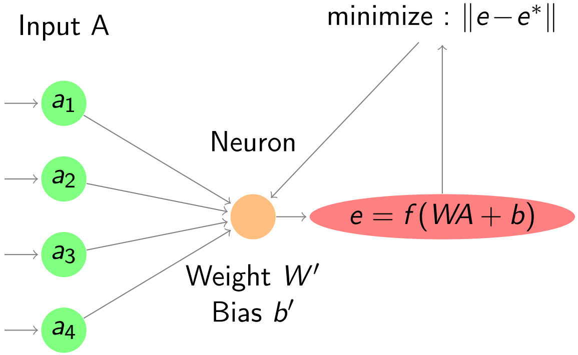
LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998d). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278–2324.

Formal neuron



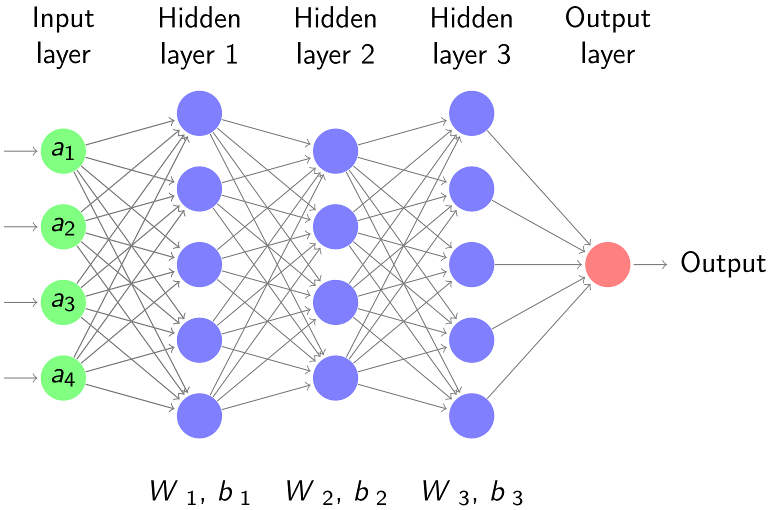
LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998d). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278–2324.

Formal neuron

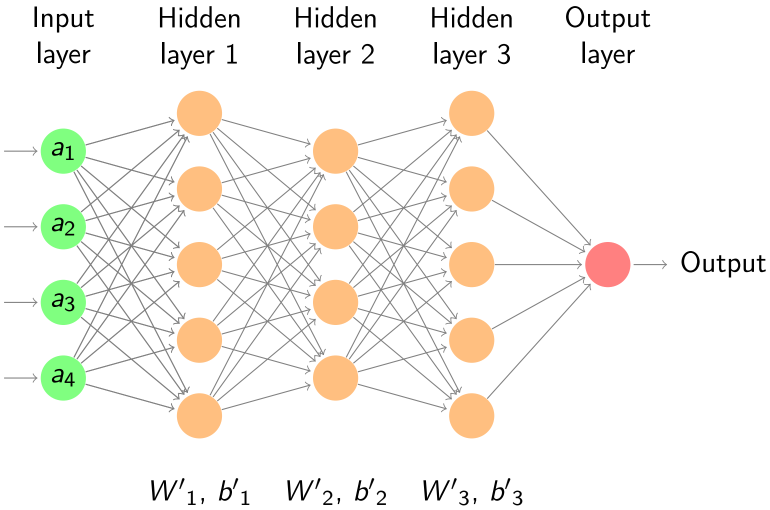


LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998d). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278–2324.

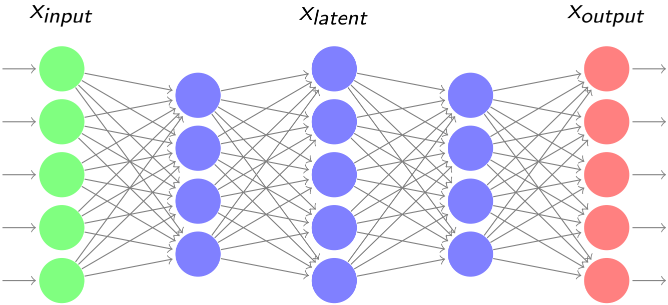
Neural network



Neural network

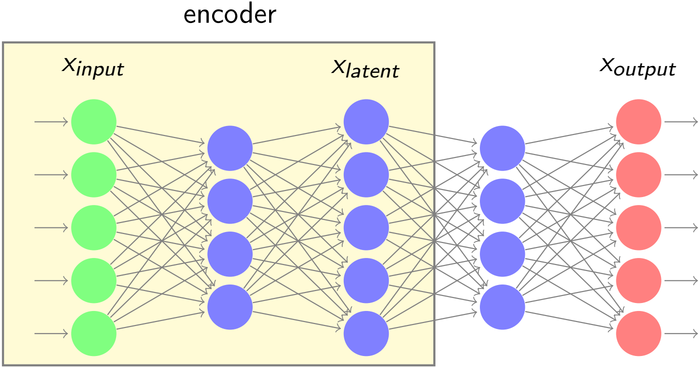


Autoencoder



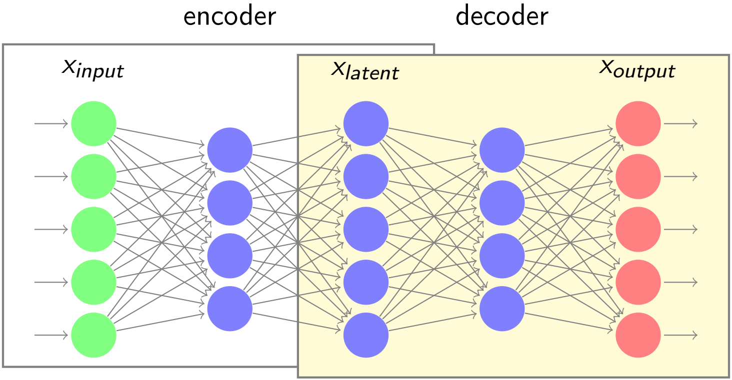
G.E. Hinton and R.R. Salakhutdinov, Reducing the Dimensionality of Data with Neural Networks, Science, 28 July 2006, Vol. 313. no. 5786, pp. 504 - 507

Autoencoder



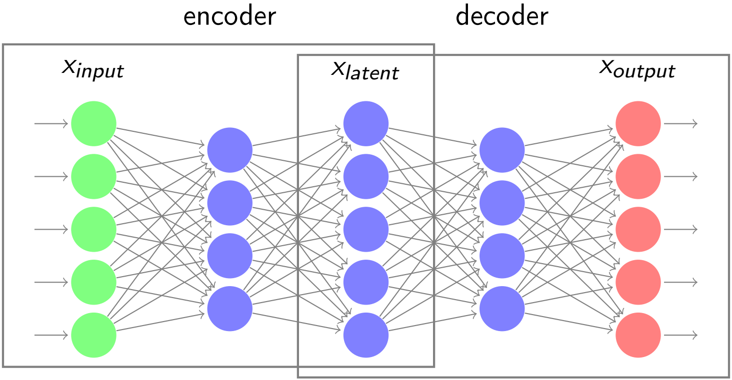
G.E. Hinton and R.R. Salakhutdinov, Reducing the Dimensionality of Data with Neural Networks, Science, 28 July 2006, Vol. 313. no. 5786, pp. 504 - 507

Autoencoder



G.E. Hinton and R.R. Salakhutdinov, Reducing the Dimensionality of Data with Neural Networks, Science, 28 July 2006, Vol. 313. no. 5786, pp. 504 - 507

Autoencoder



$$x_{latent} = encoder(x_{input})$$
$$x_{output} = decoder(x_{latent}) \simeq x_{input}$$

G.E. Hinton and R.R. Salakhutdinov, Reducing the Dimensionality of Data with Neural Networks, Science, 28 July 2006, Vol. 313. no. 5786, pp. 504 - 507

Autoencoder

Danger : Learning the identity

1. G.E. Hinton and R.R. Salakhutdinov, Reducing the Dimensionality of Data with Neural Networks, Science, 28 July 2006, Vol. 313. no. 5786, pp. 504 - 507

2. P.Vincent, H. Larochelle Y. Bengio and P.A. Manzagol, Extracting and Composing Robust Features with Denoising Autoencoders, Proceedings of the Twenty-fifth International Conference on Machine Learning (ICML'08), pages 1096 - 1103, ACM, 2008.

Autoencoder

Danger : Learning the identity

Several solutions :

1. G.E. Hinton and R.R. Salakhutdinov, Reducing the Dimensionality of Data with Neural Networks, Science, 28 July 2006, Vol. 313. no. 5786, pp. 504 - 507

2. P.Vincent, H. Larochelle Y. Bengio and P.A. Manzagol, Extracting and Composing Robust Features with Denoising Autoencoders, Proceedings of the Twenty-fifth International Conference on Machine Learning (ICML'08), pages 1096 - 1103, ACM, 2008.

Autoencoder

Danger : Learning the identity

Several solutions :

Compressing [1] :
$size(x_{latent}) < size(x_{input})$

1. G.E. Hinton and R.R. Salakhutdinov, Reducing the Dimensionality of Data with Neural Networks, Science, 28 July 2006, Vol. 313. no. 5786, pp. 504 - 507

2. P.Vincent, H. Larochelle Y. Bengio and P.A. Manzagol, Extracting and Composing Robust Features with Denoising Autoencoders, Proceedings of the Twenty-fifth International Conference on Machine Learning (ICML'08), pages 1096 - 1103, ACM, 2008.

Autoencoder

Danger : Learning the identity

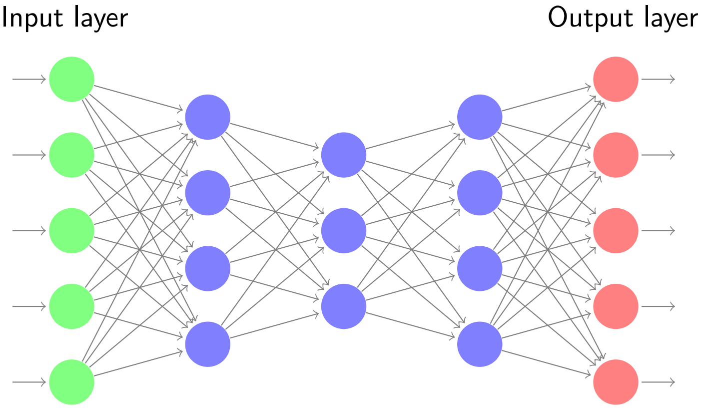
Several solutions :

Compressing[1] :
$size(x_{latent}) < size(x_{input})$

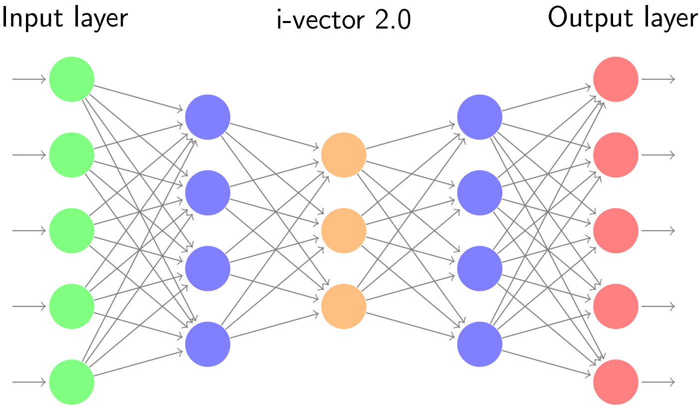
Adding noise[2] :
$x_{input} = objective + noise$

1. G.E. Hinton and R.R. Salakhutdinov, Reducing the Dimensionality of Data with Neural Networks, Science, 28 July 2006, Vol. 313. no. 5786, pp. 504 - 507
2. P.Vincent, H. Larochelle Y. Bengio and P.A. Manzagol, Extracting and Composing Robust Features with Denoising Autoencoders, Proceedings of the Twenty-fifth International Conference on Machine Learning (ICML'08), pages 1096 - 1103, ACM, 2008.

New representation



New representation



Outline

Filtering out non-speaker noise

- Filter out non-speaker dependant features

$$M = m + Tw$$

Filtering out non-speaker noise

- Filter out non-speaker dependant features (noise)
- Need to denoise the signal

$$M = noise + s_{speaker}$$

Filtering out non-speaker noise

- Filter out non-speaker dependant features (noise)
- Need to denoise the signal
- Same speaker, different signals
- Same signal, different non-speaker dependant noise

$$M_1 = noise_1 + s_{speaker}$$
$$M_2 = noise_2 + s_{speaker}$$
$$s_{speaker} = encode(M)$$

Processed data

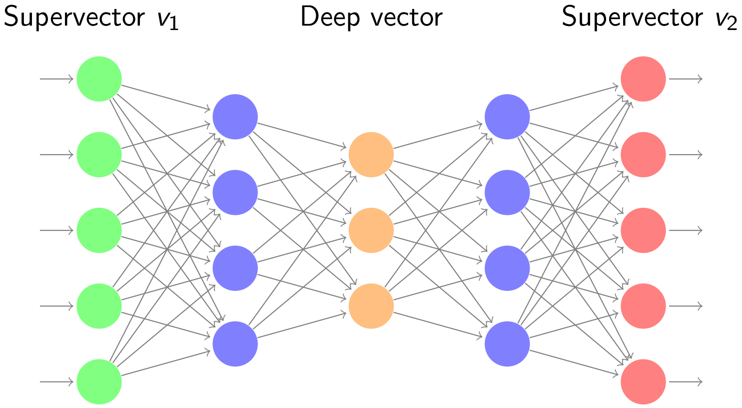
- **Raw data :** 15308 numeric sound files from BFMTV with labeled speakers
- **Pre-processed data :** 3 678 470 pairs (v_1, v_2) of supervectors spoken by the same person
- **Input :** Supervector v_1 of length 2304
- **Output :** Supervector v_2 of length 2304

Processed data

- **Raw data** : 15308 numeric sound files from BFMTV with labeled speakers
- **Pre-processed data** : 3 678 470 pairs (v_1, v_2) of supervectors spoken by the same person
- **Input** : Supervector v_1 of length 2304
- **Output** : Supervector v_2 of length 2304

$$\begin{bmatrix} v_1^{0,0} \\ v_1^{0,1} \\ \dots \\ v_1^{0,63} \\ v_1^{1,0} \\ .. \\ v_1^{N,63} \end{bmatrix} \begin{bmatrix} v_2^{0,0} \\ v_2^{0,1} \\ \dots \\ v_2^{0,63} \\ v_2^{1,0} \\ .. \\ v_2^{N,63} \end{bmatrix}$$

New representation



Intermediate vector evaluation

Preliminary evaluation with cosine similarity

Threshold t

distance $\leq t$
same speaker

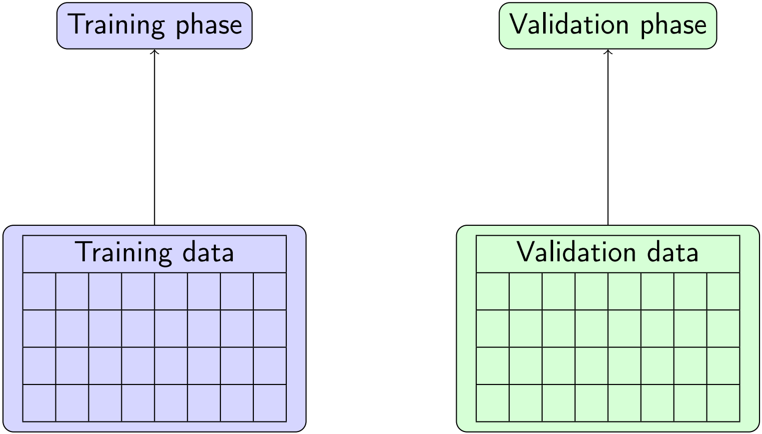
distance $> t$
different speakers

Dataset

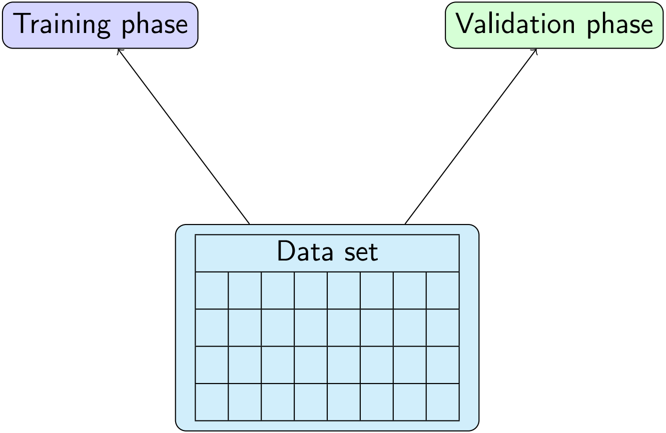
Training phase

Validation phase

Dataset

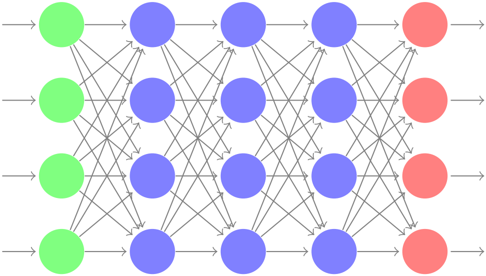

Training phase
Validation phase
Training data
Validation data

Dataset



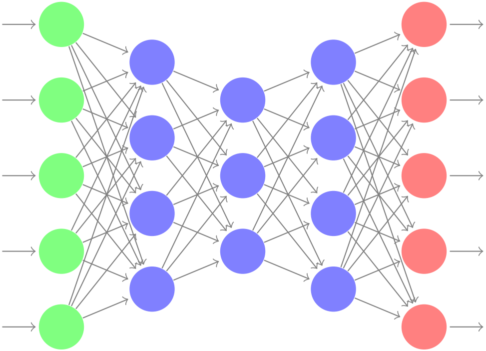
Hyper-parameters

- Number of layer



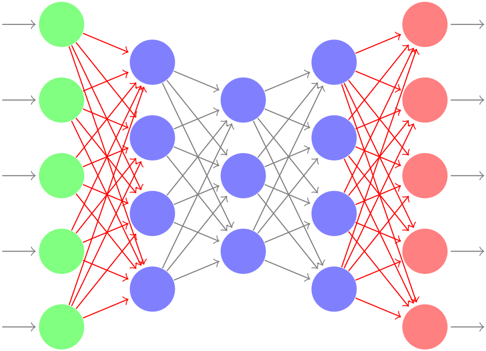
Hyper-parameters

- Number of layer
- Size of the layers



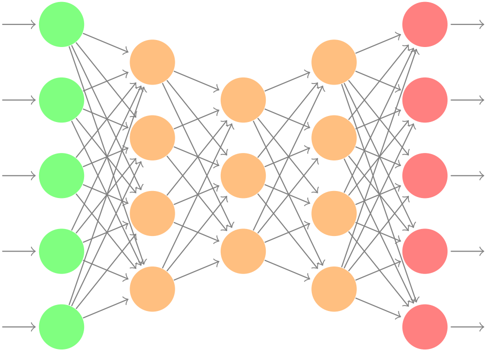
Hyper-parameters

- Number of layer
- Size of the layers
- Tied weights



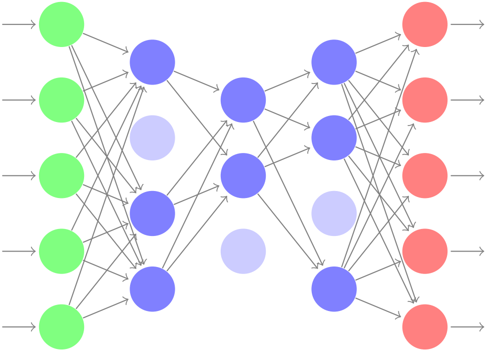
Hyper-parameters

- Number of layer
- Size of the layers
- Tied weights
- Optimizer



Hyper-parameters

- Number of layer
- Size of the layers
- Tied weights
- Optimizer
- Dropout



Outline

Repartition histograms

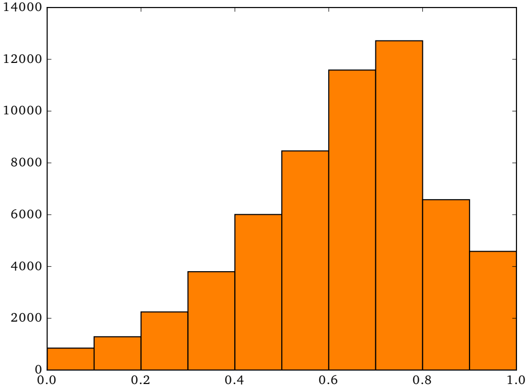


Figure – Repartition of the cosine distance between deep vectors from the same speaker

Repartition histograms

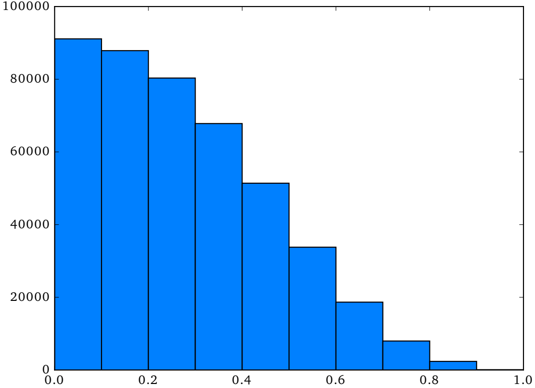


Figure – Repartition of the cosine distance between deep vectors from different speakers

Repartition histograms

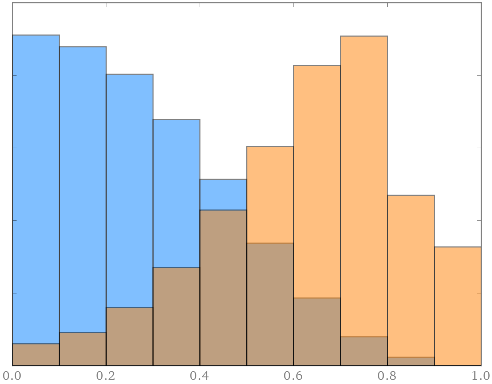


Figure – Repartition of the cosine distance between deep vectors

t-Distributed Stochastic Neighbor Embedding (t-SNE)

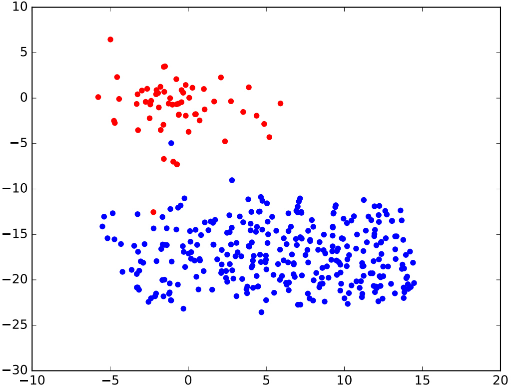
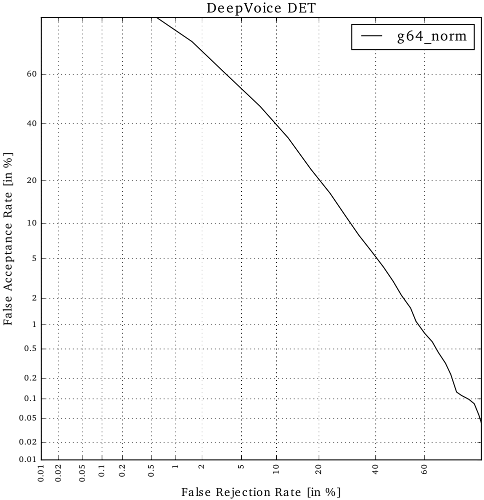
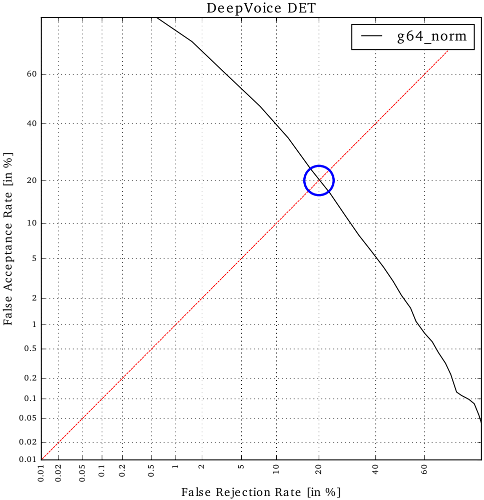


Figure – t-SNE of the deep vectors of two different speakers

Detection Error Tradeoff (DET) graph

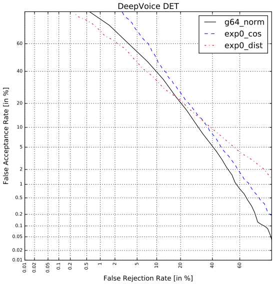


Detection Error Tradeoff (DET) graph



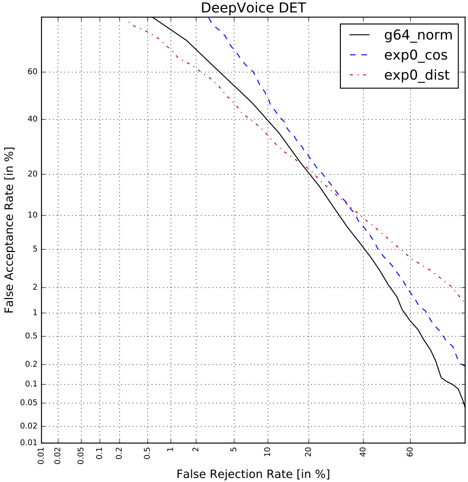
Experiment 0

Number of layers	5				
Size of layers	2304	1000	50	10000	2304
Tied weights	No				
Optimizer	Gradient Descent				
Dropout	0.90				



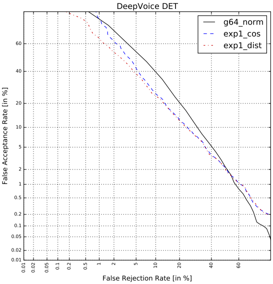
Experiment 0

Number of layers	5				
Size of layers	2304	1000	50	10000	2304
Tied weights	No				
Optimizer	Gradient Descent				
Dropout	0.90				



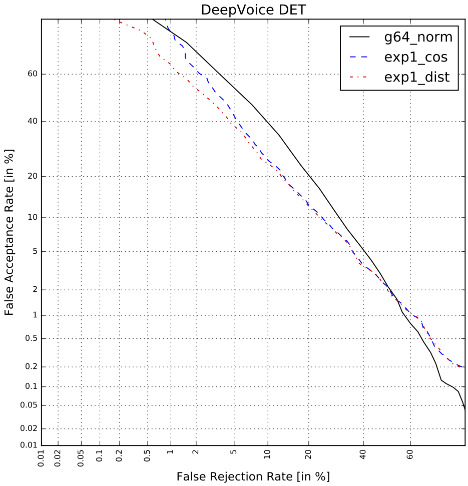
Experiment 1

Number of layers	5				
Size of layers	2304	1000	50	10000	2304
Tied weights	No				
Optimizer	Adam				
Dropout	0.90				



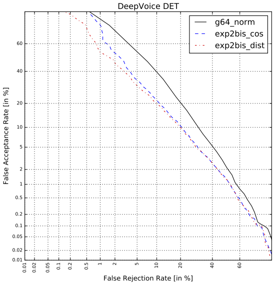
Experiment 1

Number of layers	5				
Size of layers	2304	1000	50	10000	2304
Tied weights	No				
Optimizer	Adam				
Dropout	0.90				



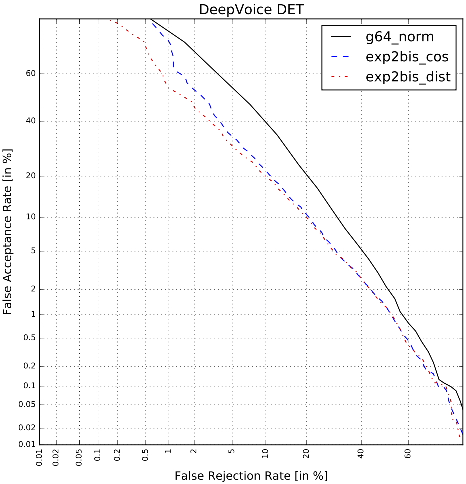
Experiment 2

Number of layers	5				
Size of layers	2304	480	100	480	2304
Tied weights	No				
Optimizer	Adam				
Dropout	0.90				



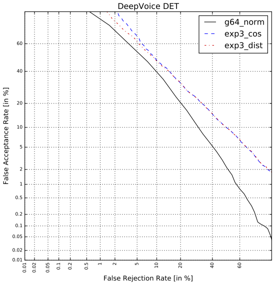
Experiment 2

Number of layers	5				
Size of layers	2304	480	100	480	2304
Tied weights	No				
Optimizer	Adam				
Dropout	0.90				



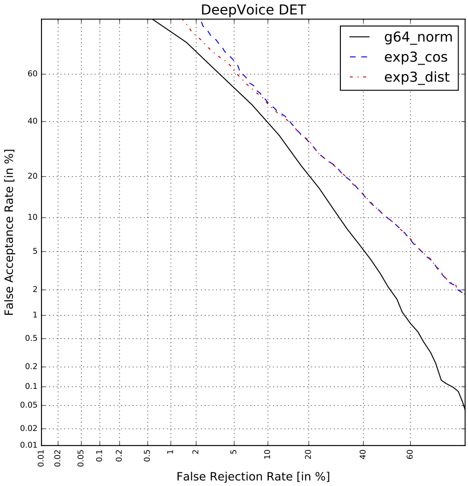
Experiment 3

Number of layers	7						
Size of layers	2304	720	225	70	225	720	2304
Tied weights	No						
Optimizer	Adam						
Dropout	0.90						



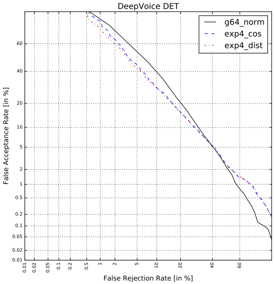
Experiment 3

Number of layers	7						
Size of layers	2304	720	225	70	225	720	2304
Tied weights	No						
Optimizer	Adam						
Dropout	0.90						



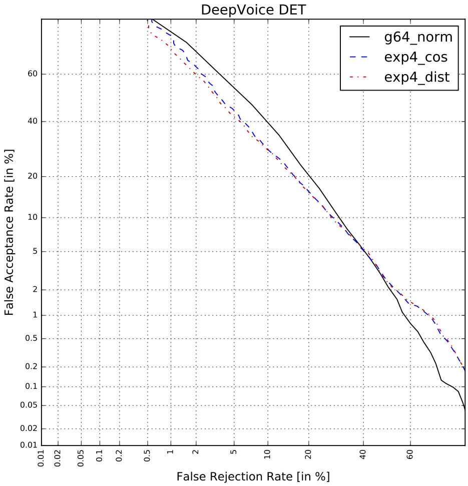
Experiment 4

Number of layers	7						
Size of layers	2304	720	225	70	225	720	2304
Tied weights	Yes						
Optimizer	Adam						
Dropout	0.90						



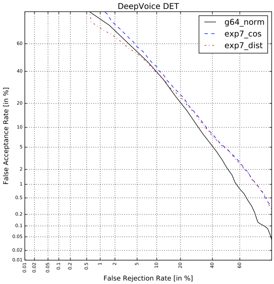
Experiment 4

Number of layers	7						
Size of layers	2304	720	225	70	225	720	2304
Tied weights	Yes						
Optimizer	Adam						
Dropout	0.90						



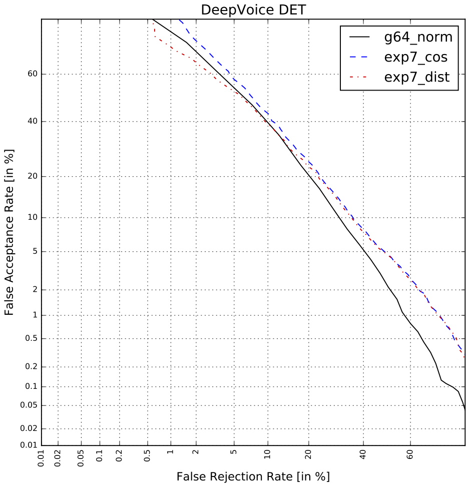
Experiment 7

Number of layers	5				
Size of layers	2304	500	80	500	2304
Tied weights	No				
Optimizer	Adam				
Dropout	0.80				



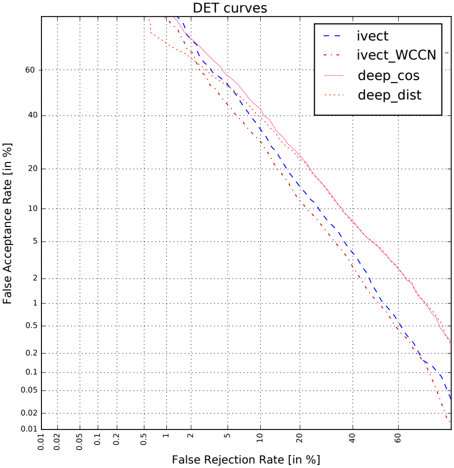
Experiment 7

Number of layers	5				
Size of layers	2304	500	80	500	2304
Tied weights	No				
Optimizer	Adam				
Dropout	0.80				



Outline

Deep-vectors vs. i-vectors



Further work

- Run additional experiments
- Adjust the hyper-parameters
- Run more experiments with disjoint training set and evaluation set