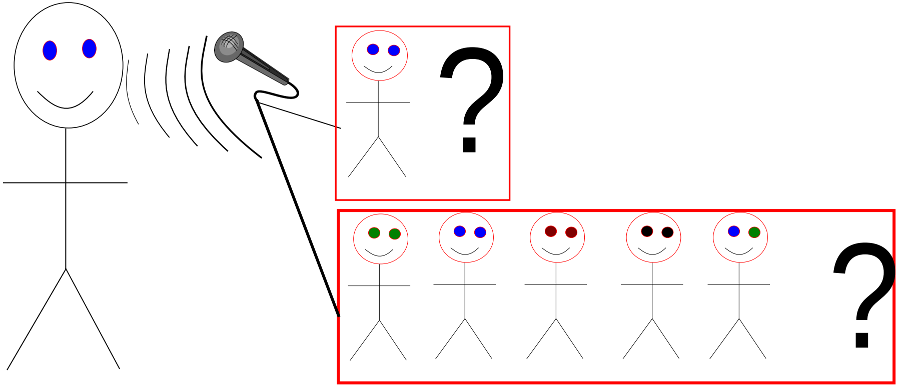
DeepVoice

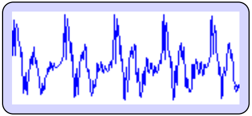
Extracting meaningful signal representation for Speaker Recognition using deep architectures

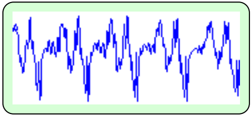
Rémi Hutin, Raphaël Truffet
Supervisors : Guillaume Gravier and Vedran Vukotić

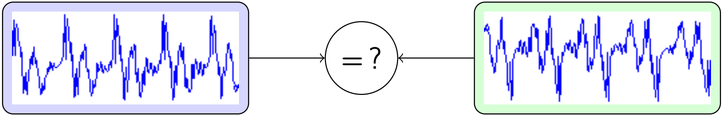
Computer science department
ENS Rennes

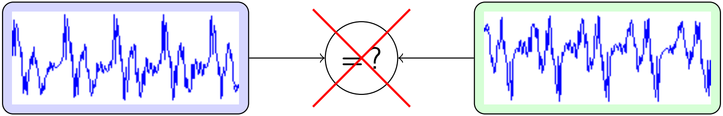
Linkmedia project
IRISA

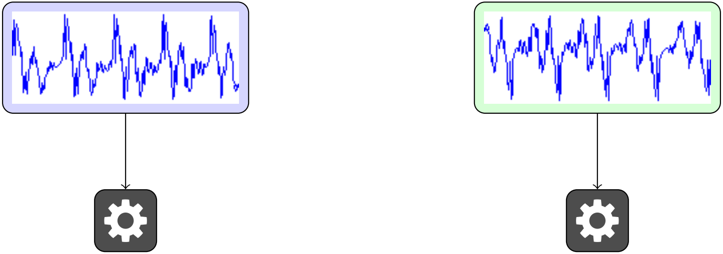


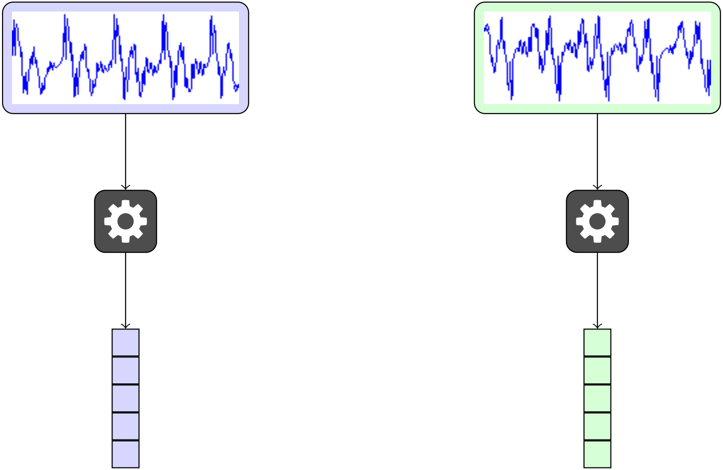


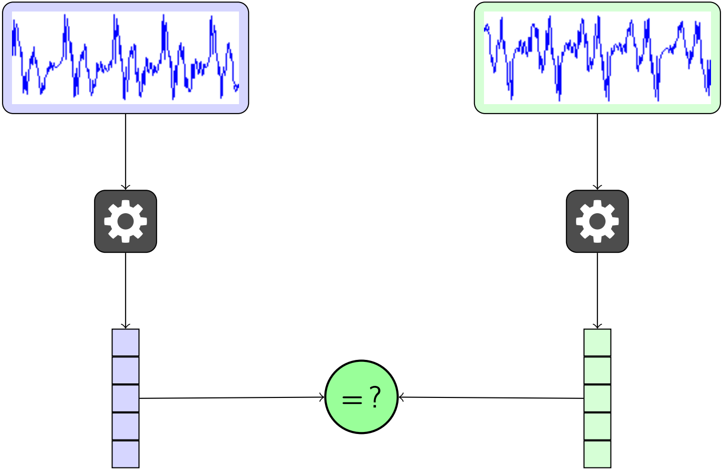


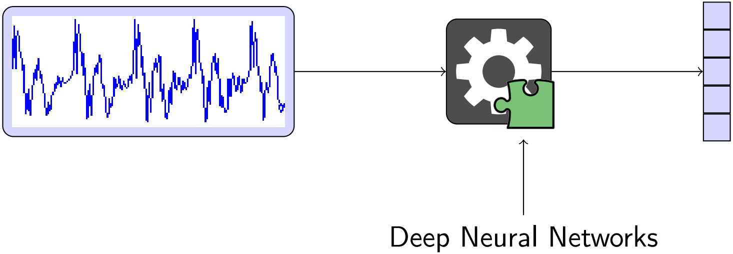








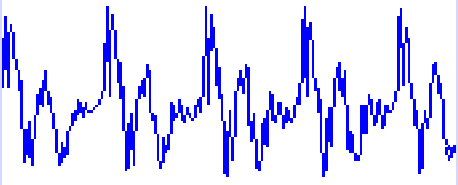



Deep Neural Networks

Outline

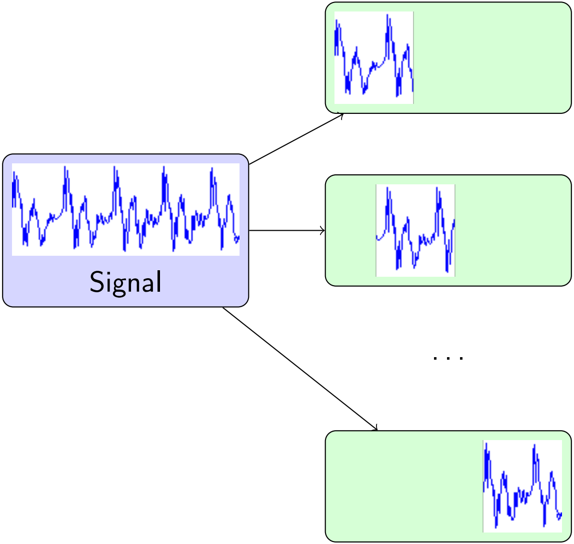
Outline

Signal processing workflow

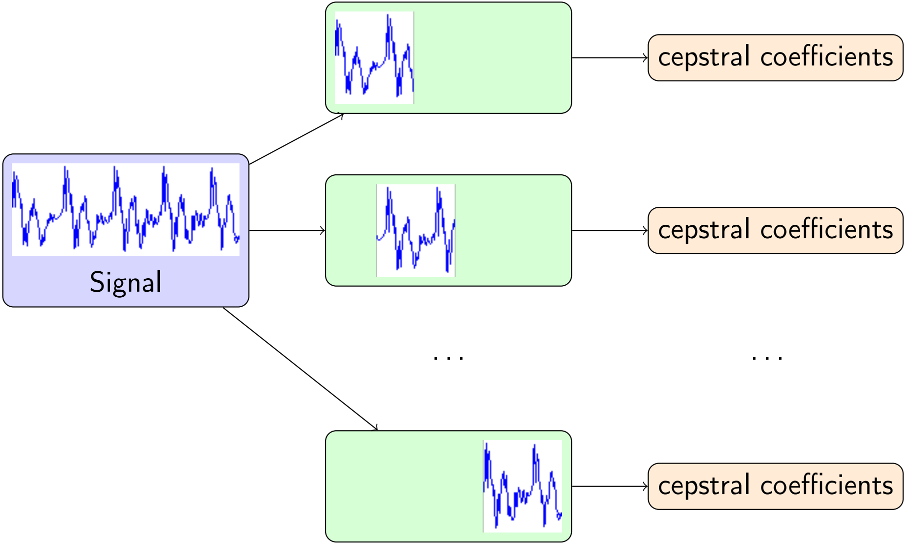


Signal

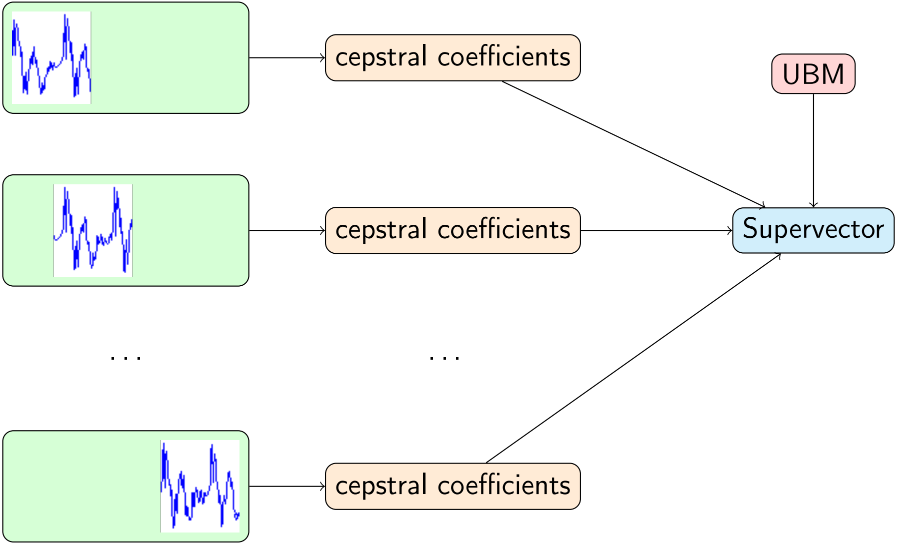
Signal processing workflow



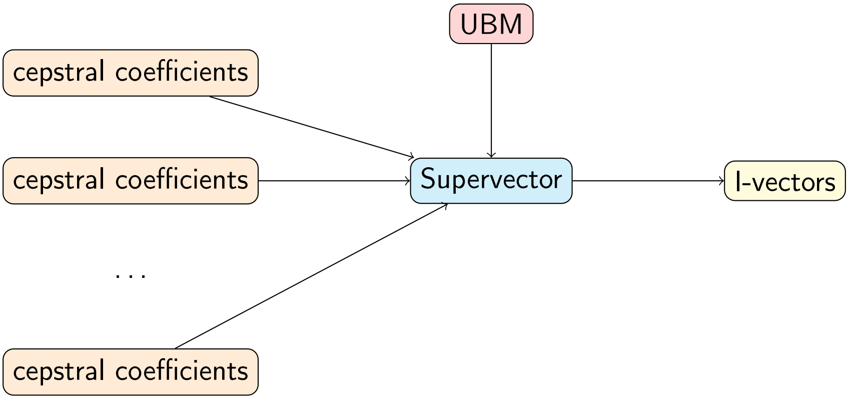
Signal processing workflow



Signal processing workflow



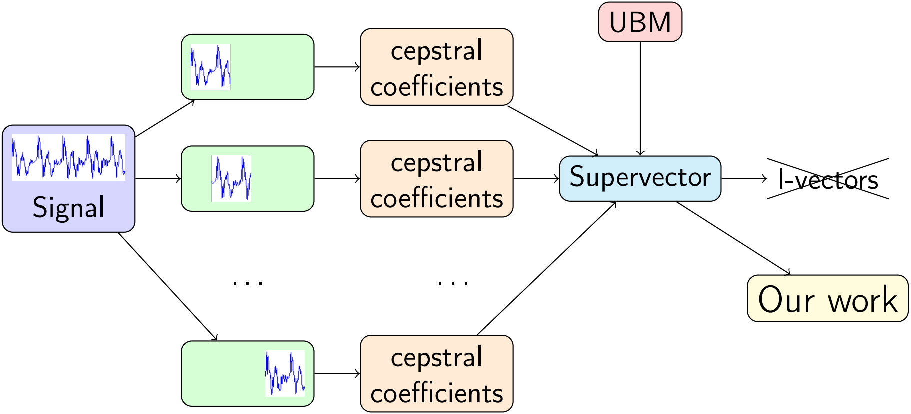
Signal processing workflow



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 :

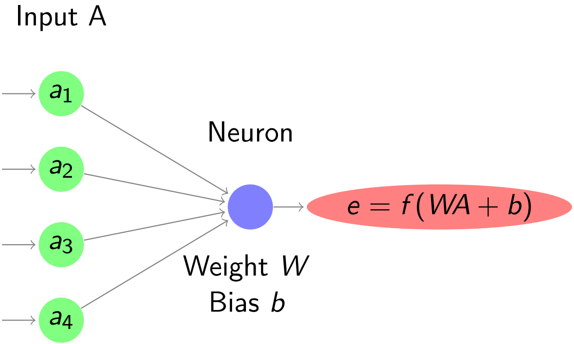
- **Non-linear** feature extraction
- They naturally generate several level of representation
- They bring out unsuspected features

Use of Deep Neural Networks (DNN)

Deep neural networks are interesting because :

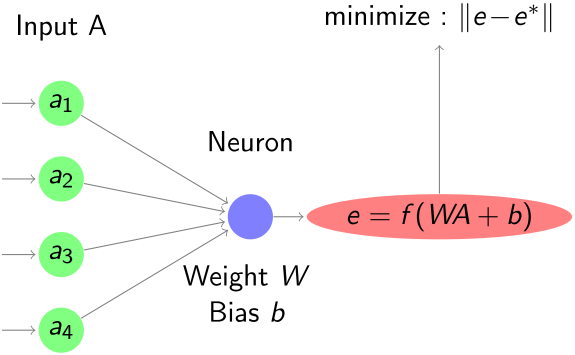
- **Non-linear** feature extraction
- They naturally generate several level of representation
- They bring out unsuspected features
- There is a multitude of architectures

Formal neuron



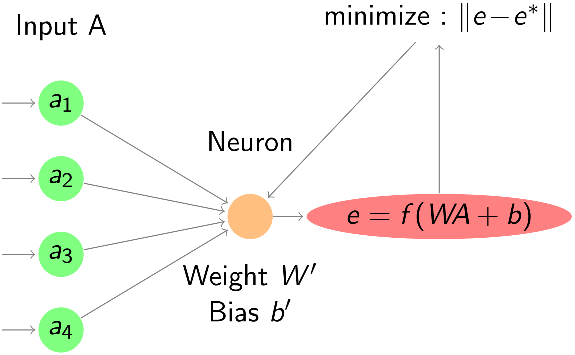
LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998d). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278–2324.

Formal neuron



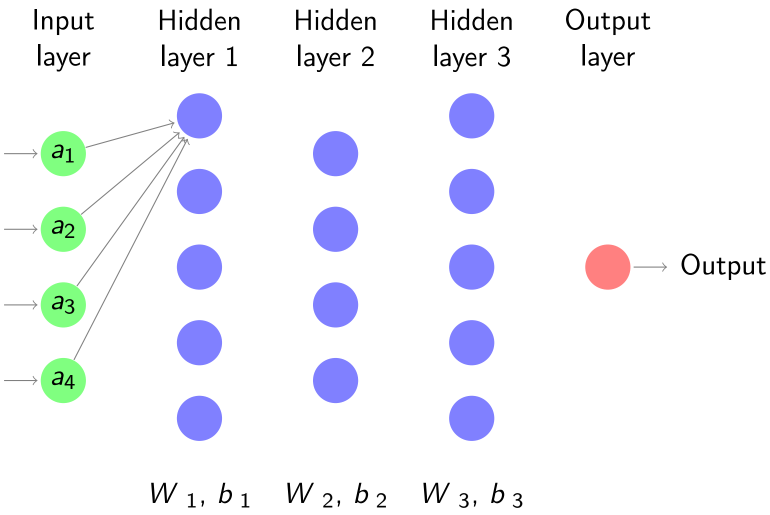
LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998d). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278–2324.

Formal neuron

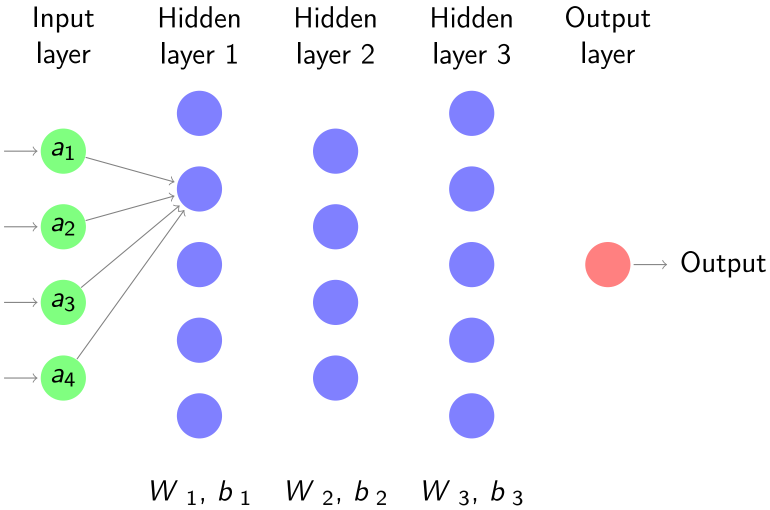


LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998d). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278–2324.

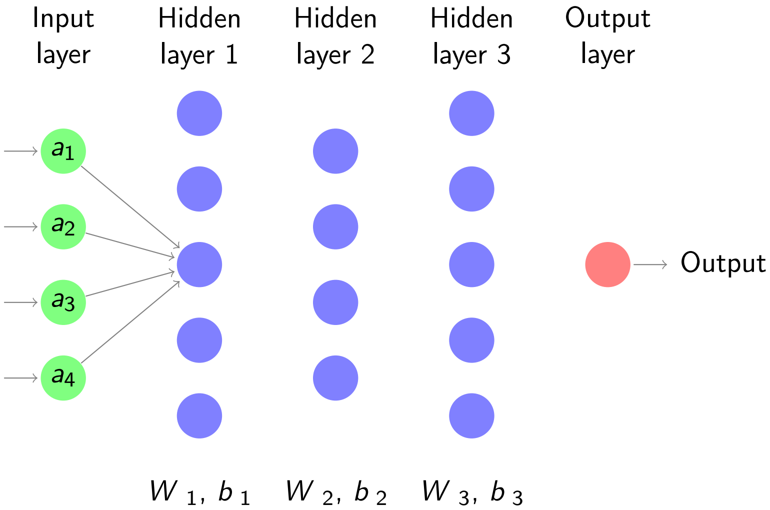
Neural network



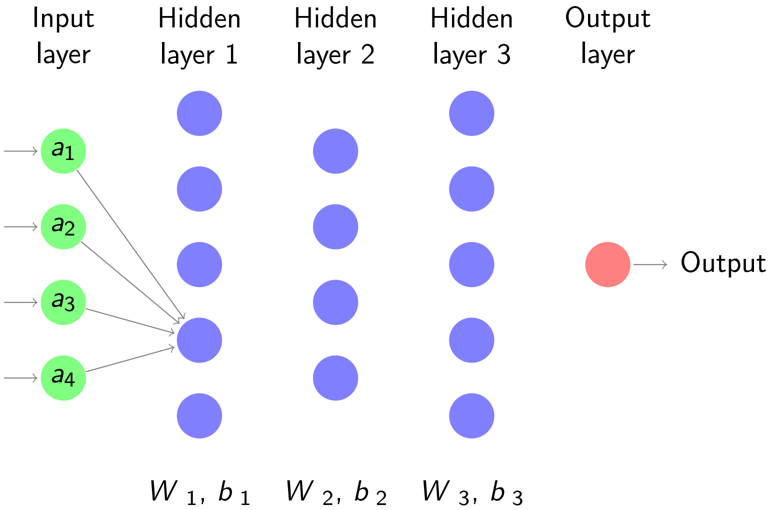
Neural network



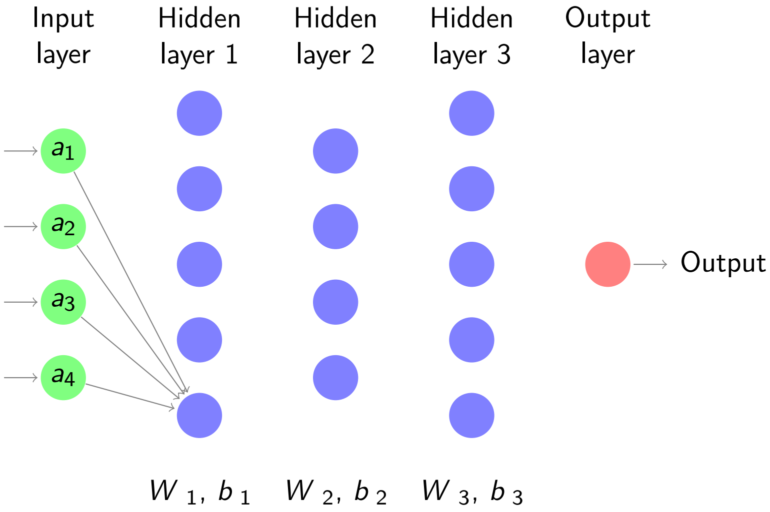
Neural network



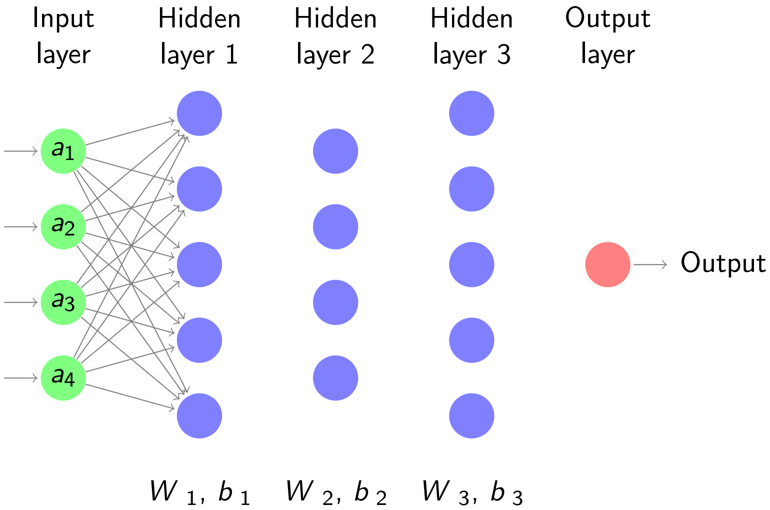
Neural network



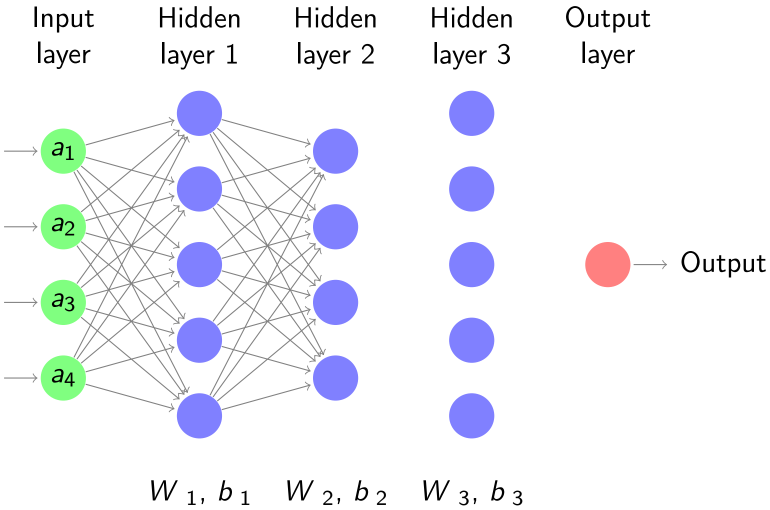
Neural network



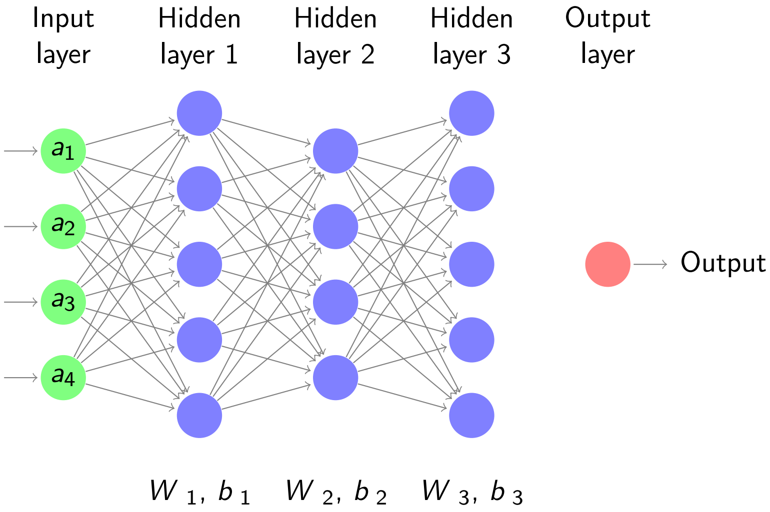
Neural network



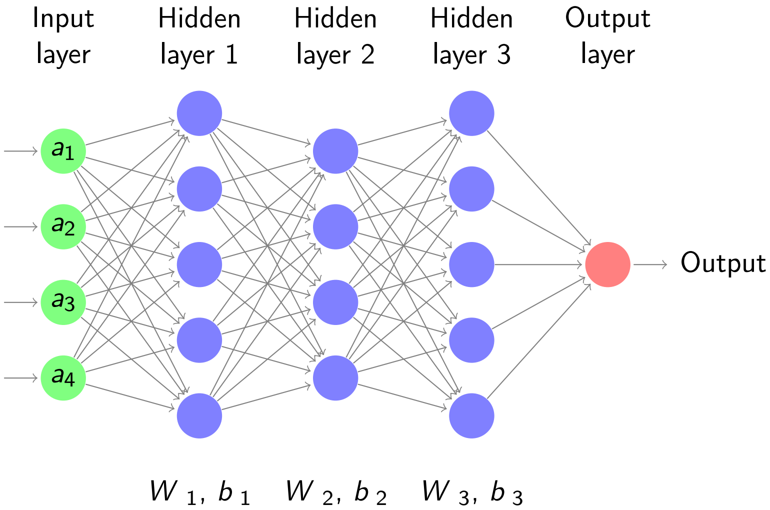
Neural network



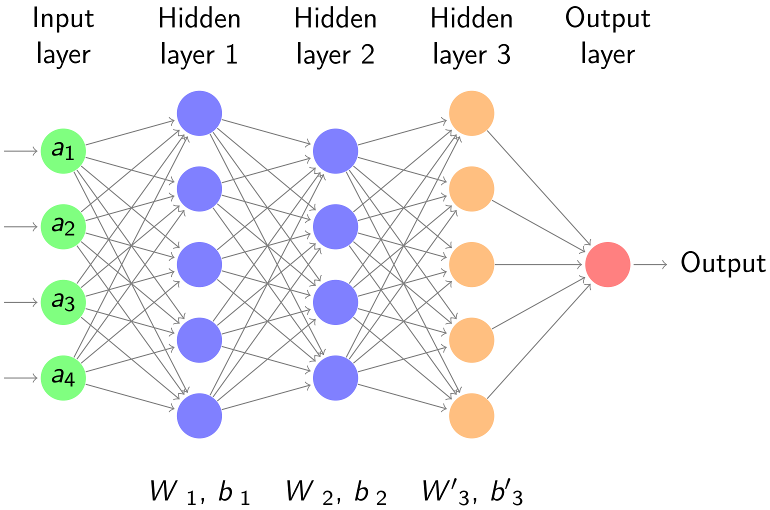
Neural network



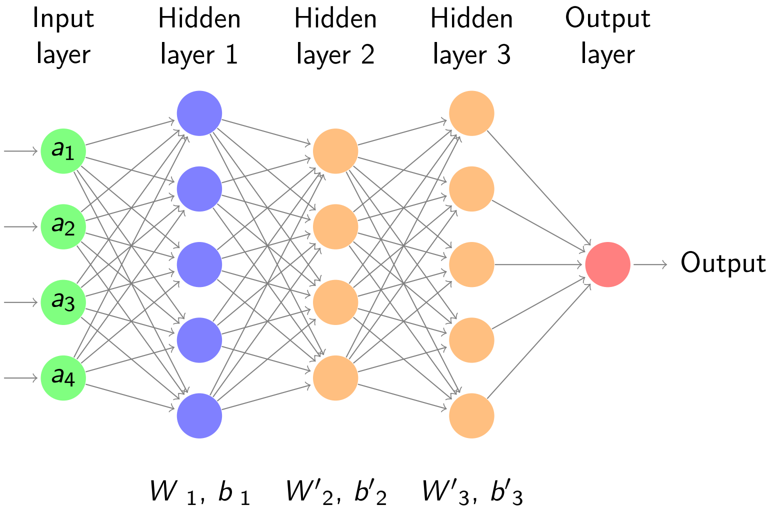
Neural network



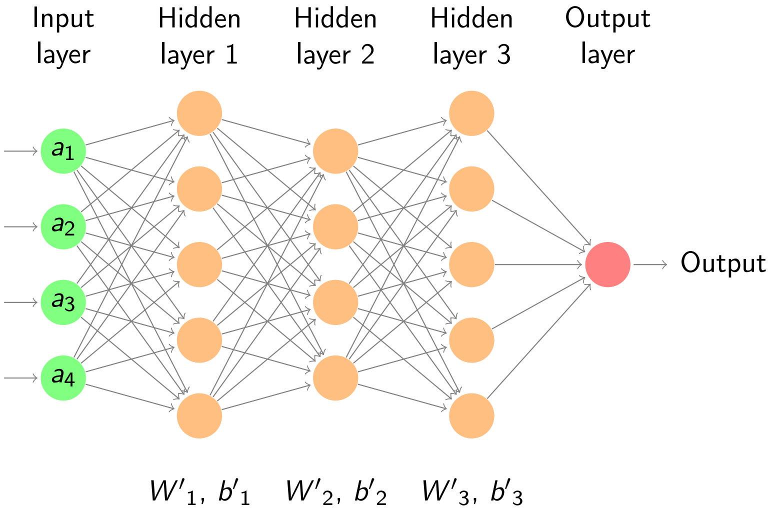
Neural network



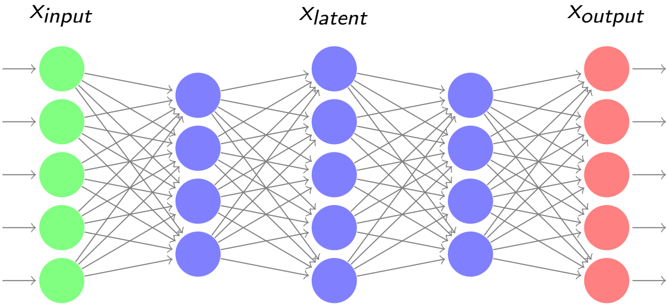
Neural network



Neural network

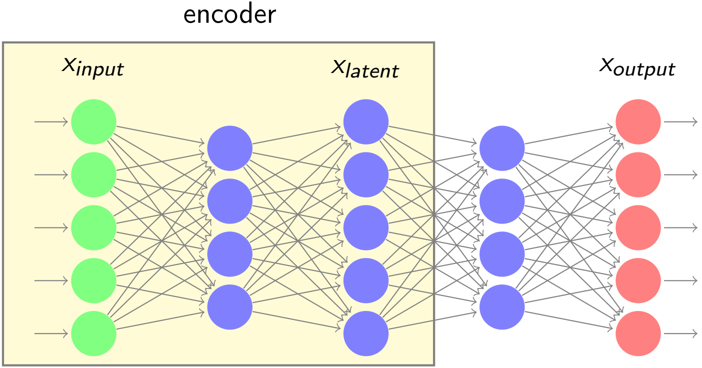


Autoencoder



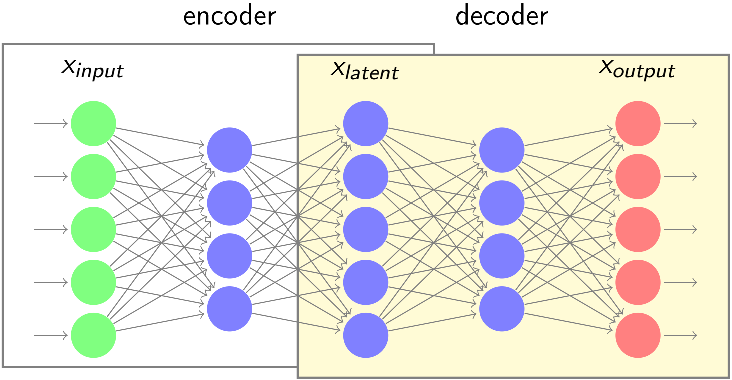
G.E. Hinton and R.R. Salakhutdinov, Reducing the Dimensionality of Data with Neural Networks, Science, 28 July 2006, Vol. 313. no. 5786, pp. 504 - 507

Autoencoder



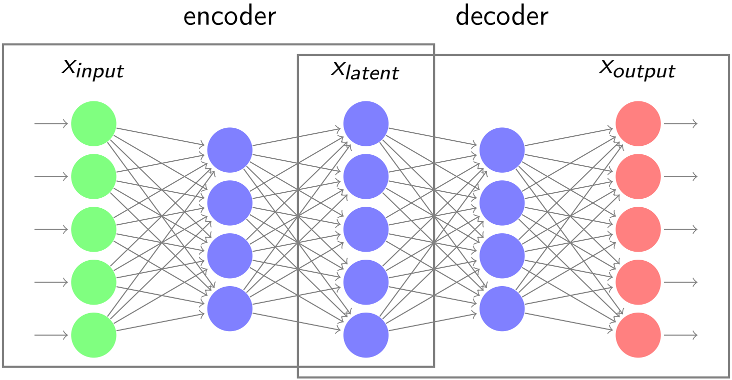
G.E. Hinton and R.R. Salakhutdinov, Reducing the Dimensionality of Data with Neural Networks, Science, 28 July 2006, Vol. 313. no. 5786, pp. 504 - 507

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 :

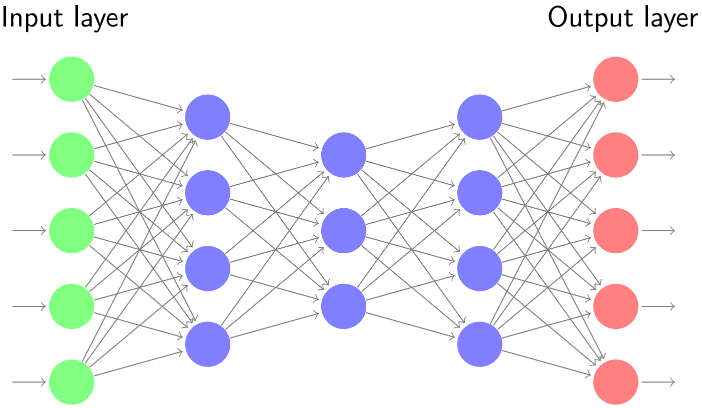
Compressing [1] :
$size(x_{latent}) < size(x_{input})$

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

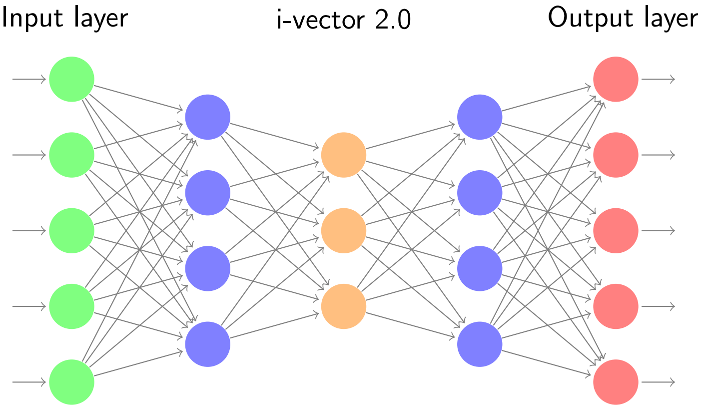
1. G.E. Hinton and R.R. Salakhutdinov, Reducing the Dimensionality of Data with Neural Networks, Science, 28 July 2006, Vol. 313. no. 5786, pp. 504 - 507

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

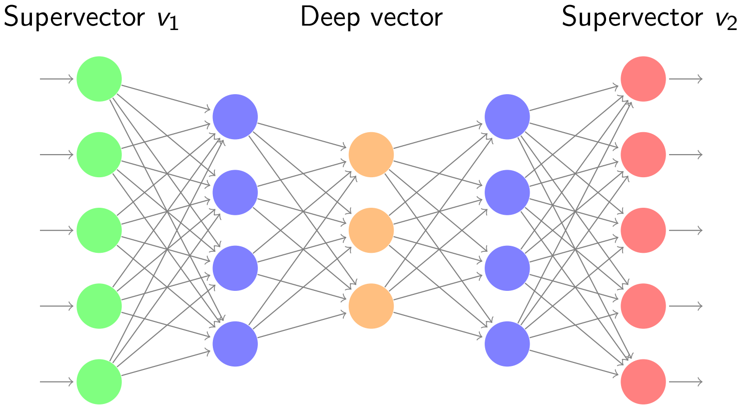
- **Raw data :** 15311 numeric sound files from BFMTV with labeled speakers
- **Pre-processed data :** 3 678 470 pairs (v_1, v_2) of supervectors spoken by the same person
- **Input :** Supervector v_1 of length 2304
- **Output :** Supervector v_2 of length 2304

Processed data

- **Raw data** : 15311 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

$$\begin{bmatrix} v_1^{0,0} \\ v_1^{0,1} \\ \cdots \\ v_1^{0,255} \\ v_1^{1,0} \\ .. \\ v_1^{N,255} \end{bmatrix} \begin{bmatrix} v_2^{0,0} \\ v_2^{0,1} \\ \cdots \\ v_2^{0,255} \\ v_2^{1,0} \\ .. \\ v_2^{N,255} \end{bmatrix}$$

New representation



Intermediate vector evaluation

Preliminary evaluation with cosine similarity

Threshold t

$distance \leq t$
same speaker

$distance > t$
different speakers

Outline

Goals

Numeric signals represented by i-vectors for speaker recognition tasks.
We seek to offer an alternative with deep neural networks.
What does it mean to improve on i-vectors ?

- Better compression

- Better results on angular threshold

- State-of-the-art results for speaker recognition

Goals and expected issues

Numeric signals represented by i-vectors for speaker recognition tasks.
We seek to offer an alternative with deep neural networks.
What does it mean to improve on i-vectors ?

- Better compression
  - Compression size
  - Hyperparameters
  - Compromise with results
- Better results on angular threshold
  - Optimization method
  - Compromise with compression
- State-of-the-art results for speaker recognition
  - Different training sets
  - More complicated evaluation methods