Robust and efficient optimization scheme leading to KL transform

Oleksandr Pankiv, Dariusz Puchala
and Kamil Stokfiszewski
Institute of Information Technology
Lodz University of Technology
ul. Wolczanska 215, 90-924 Lodz, Poland

oleksandr.pankiv@dokt.p.lodz.pl, dariusz.puchala@p.lodz.pl,
kamil.stokfiszewski@p.lodz.pl

Data
Compression
Conference

Introduction

- Robust and efficient optimization scheme leading to KL transform
  - gradient based
  - minimization of the mean squared error of signal reconstruction
  - minimization of the entropy related criterion
- Minimal and feasible constraints
  - unit columns of coding and decoding matrices
  - feasible to uphold using standard neural optimization methods
- Neural realization
  - standard grdient based optimization
  - autoencoder neural network with two-criteria optimization
- Performance
  - leading to Karhunen-Loève transform
  - close to optimal transforms coefficients energy distribution in presence of quantization noise

General Optimization Scheme

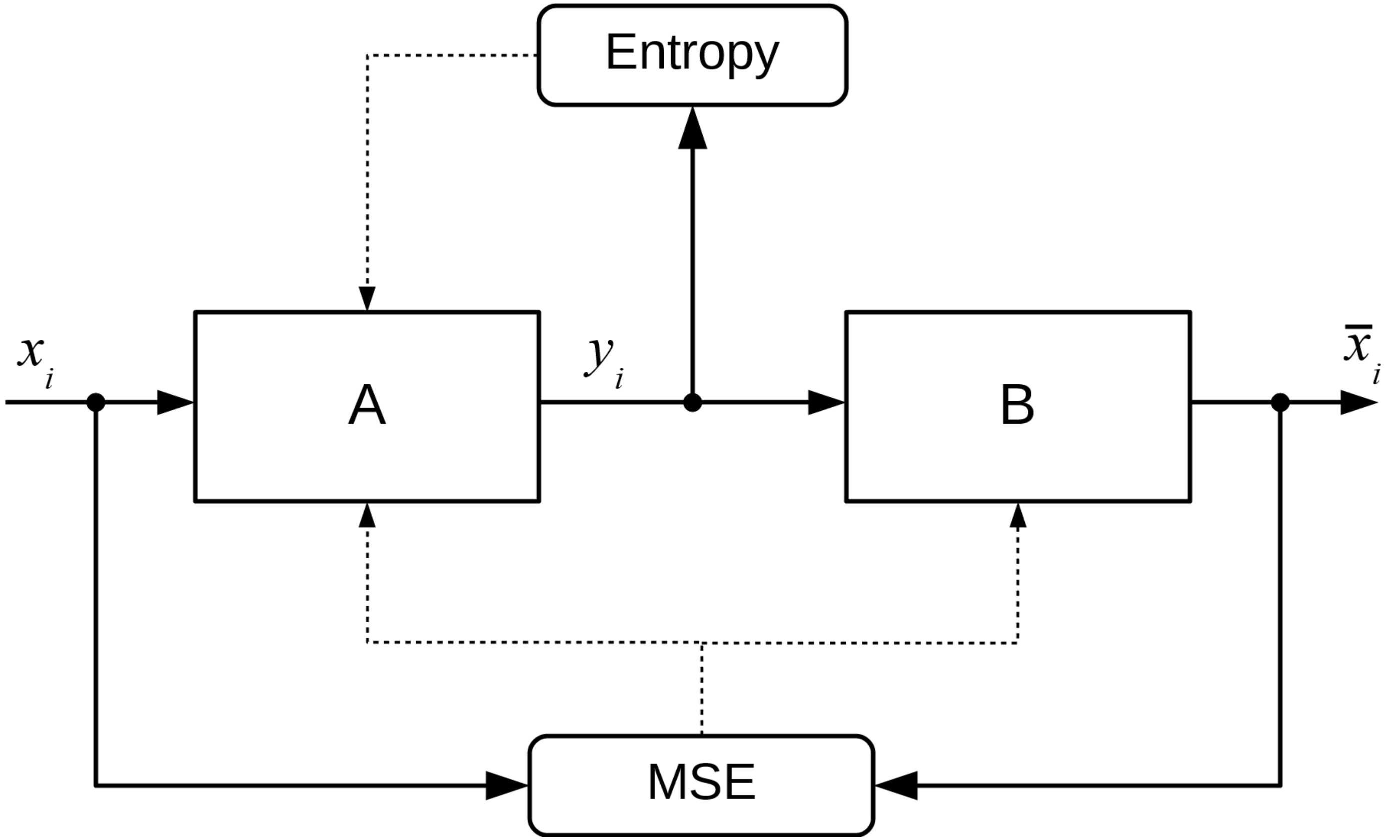


Fig. 1. The poposed general optimization scheme based on the MSE and Entropy minimization criteria.

Optimization Constraints and Criteria

- Constraints for coding and decoding matrices $\mathbf{A}$ and $\mathbf{B}$ respectively
  - $\|\mathbf{a}_i\|^2 = 1$ and $\|\mathbf{b}_i\|^2 = 1$ for $i = 0, 1, \ldots, N-1$

- Minimization criteria
  - MSE minimization: $$J_I = \frac{1}{N} \mathrm{tr}\left((\mathbf{BA} - \mathbf{I})\mathbf{R}_{\mathbf{x}}(\mathbf{BA} - \mathbf{I})^T\right)$$
  - Entropy minimization: $$\pi = \left(\prod_{n=0}^{N-1} \sigma_{y(n)}^2\right)^{1/N} \longrightarrow J_2 = \sum_{n=0}^{N-1} \log\left(\sigma_{y(n)}^2\right)$$

- Overall cost function: $$J = \alpha J_1 + \beta J_2$$

Practical Realization with Artificial Neural Networks

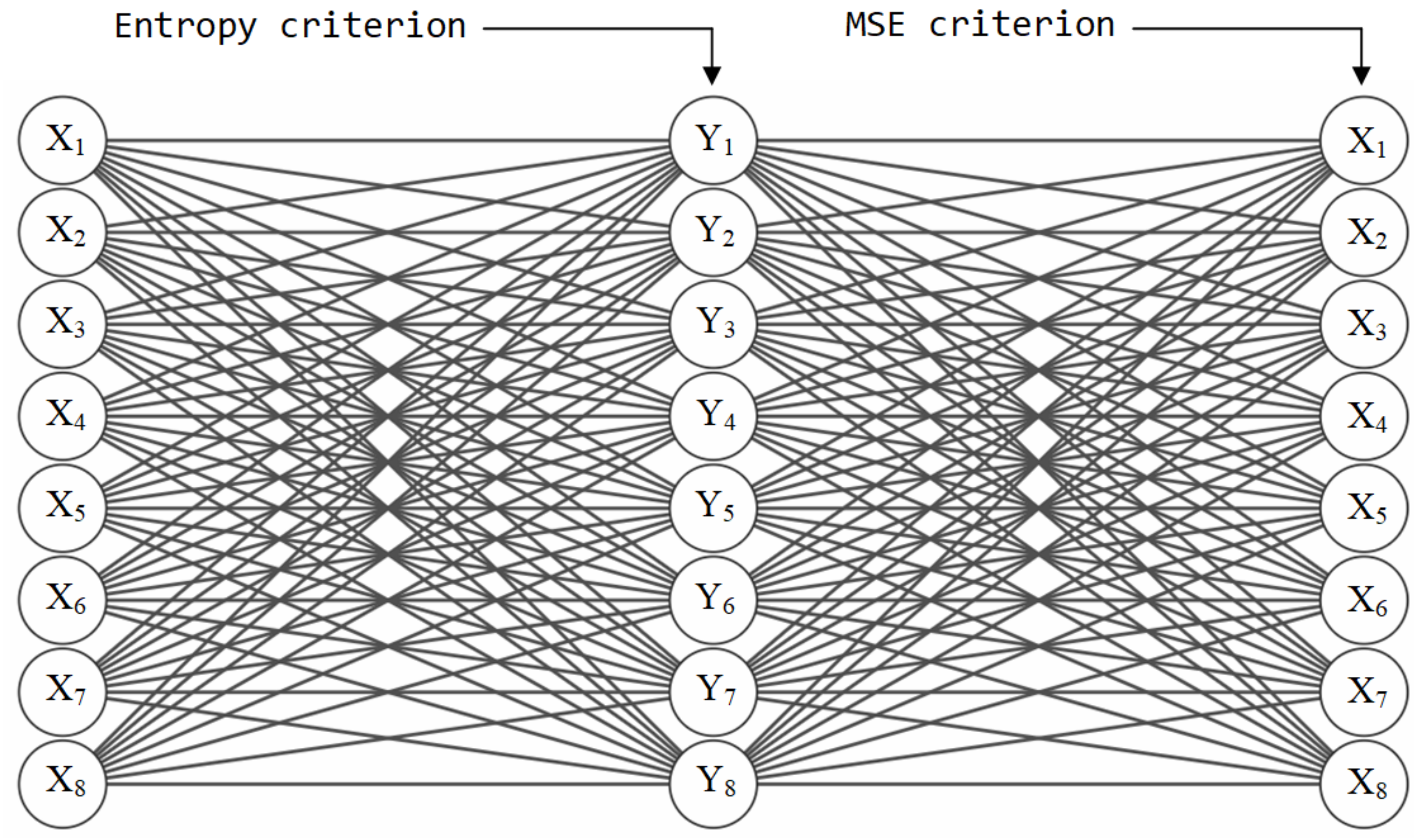


Fig. 2. The poposed general optimization scheme's neural network architecture.

```
Dense(N,activation='linear',
      use_bias=False,
      kernel_constraint=unit_norm());
```

Fig. 3a. *Keras* and *Tensorflow* ANN single layer implementation.

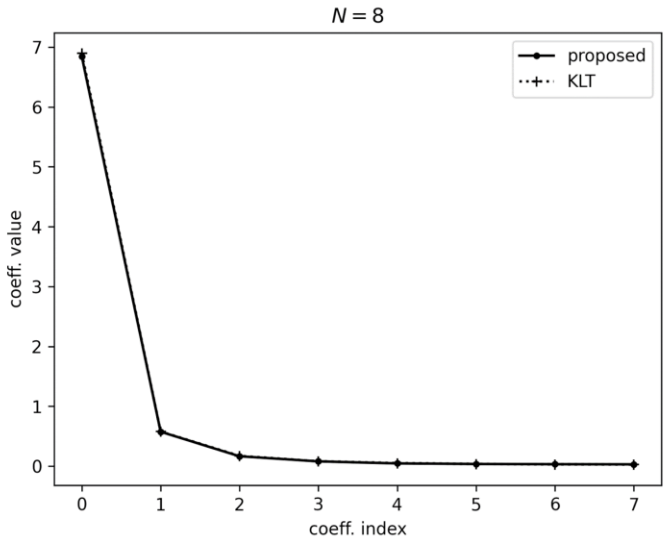
```
def J2(y,z):
    t1=tf.add(y,z);
    t2=tf.math.reduce_variance(t1,axis=1);
    t3=tf.math.log(t2);
    t4=tf.math.reduce_sum(t3);
    return t4;
```

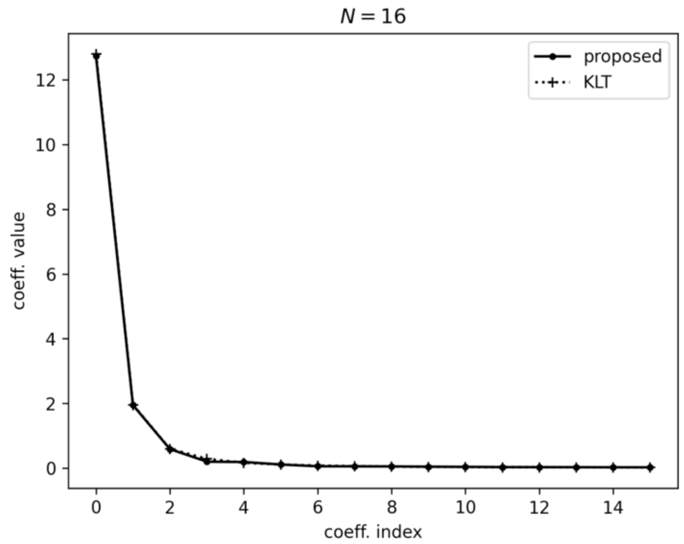
Fig. 3b. *Keras* and *Tensorflow* ANN tensor for Entropy optimization criterion.

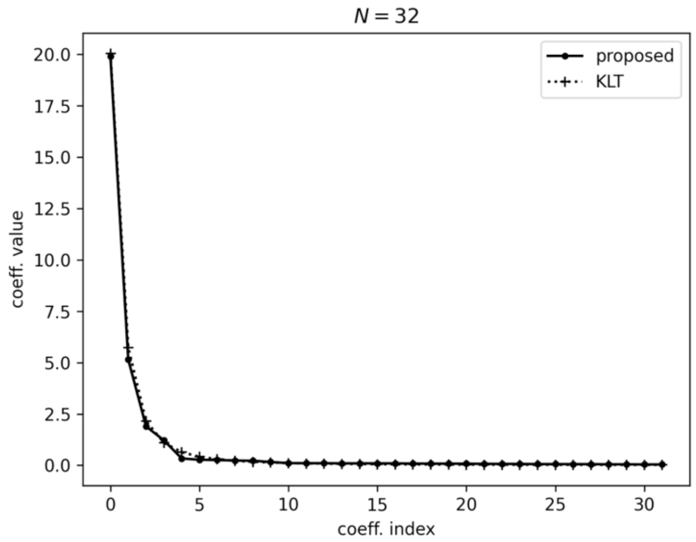
Experimental Analysis

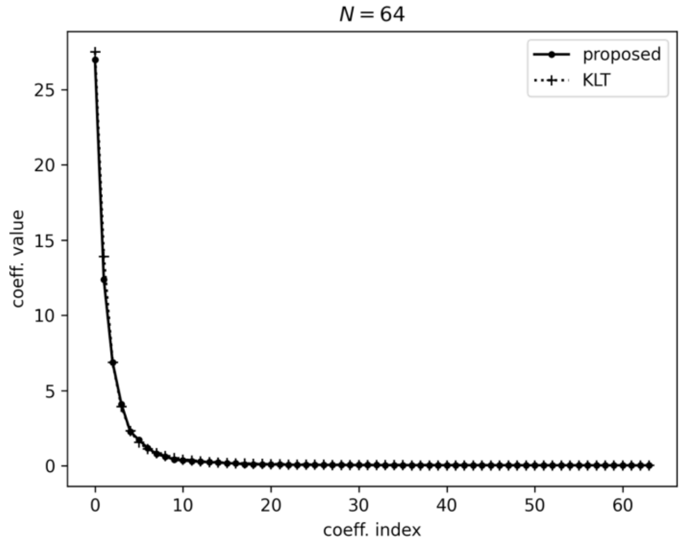
- **Conducted experiments were**
  - focused on evaluation of the effectiveness of the MSE criterion
  - focused on both criteria and aimed at evaluation of the energy distribution in the domain of $\mathbf{A}$ transform
  - focused on both criteria but with the additional stage of data quantization in the domain of $\mathbf{A}$ transform
  - performed for transform sizes 8, 16, 32 and 64 points
  - focused on orthonormlality constraint effectiveness evaluation $\epsilon_1 = \frac{1}{N^2}\|\mathbf{A}\mathbf{A}^T - \mathbf{I}\|_F^2$

Experimental Results









Figs 4. Energy distributions comparison between the obtained transforms and the KLT for transform sizes *N = 8, 16, 32* and *64* points.

- orthogonality criterion $\epsilon_1 = \frac{1}{N^2} \|\mathbf{A}\mathbf{A}^T - \mathbf{I}\|_F^2$

  for all transform sizes did not exceed $4 \cdot 10^{-5}$

Summary and Conclusions

The main conclusions regarding the presented scheme are

- Highly satisfactory results for
  - MSE optimization criterion
  - Entropy optimization criterion
  - Orthogonality constraints criterion
  - Obtained transform coefficients energy distribution criterion in comparison to KLT
- Simple linear two-stage structure of the ANN architecture
  - standard autoencoder ANN architecture
  - easy to implement using standard ANN programming libraries such as *Keras* and *Tensorflow*
- Karhunen-Loève transform equivalence
  - the obtained coding and decoding transforms follow the eigenvectors of signal autocovariance matrix
  - confirmed validity in presence of quantization noise what is crucial for practical applications

Thank You For Your Attention.