Spatio-Temporal Convolutional Sparse Auto-Encoder for Sequence Classification

Moez Baccouche, Franck Mamalet, Christian Wolf, Christophe Garcia and Atilla Baskurt

23rd British Machine Vision Conference 2012
Guildford UK. September 5th 2012
Problem statement (1)

- The dominant methodology in video sequence classification relies on so-called **hand-crafted features**

- These features are manually designed to be optimal for a specific task → **high problem dependency**:
  - Human action recognition: Harris-3D, Cuboïd, Hessian…
  - Facial expression recognition: LBP, Gabor, Haar-like…
The dominant methodology in video sequence classification relies on so-called hand-crafted features.

These features are manually designed to be optimal for a specific task → high problem dependency:

- Human action recognition: Harris-3D, Cuboïd, Hessian…
- Facial expression recognition: LBP, Gabor, Haar-like…

Another category of approaches aims to learn feature extractors instead of engineering them.

These machines are trained in two ways:

- Supervised: cf. our previous work (HBU’11)
- Unsupervised: the main goal of this work
Unsupervised learning of feature extractors mostly rely on auto-encoders:

- The encoder and the decoder are parameterized functions
- The code size is generally smaller than the input dimension
Problem statement (2)

- Unsupervised learning of feature extractors mostly rely on auto-encoders:
  - The encoder and the decoder are parameterized functions
  - The code size is generally smaller than the input dimension
  - Several recent works advocate the use of sparse over-complete representations:
    - over-complete: the code size is larger than the input dimension
    - sparse: only few values are non-zero

- The use of these models in the video case is still an open issue
Goal of this work

- Propose an auto-encoder model which automatically learns sparse over-complete spatio-temporal features

- Handle the spatial and temporal shift-invariance of the learned features

- Classify the entire sequences based on the temporal evolution of the learned features

- Provide a training procedure for the entire model

- Validate the approach by testing it on two different problems
Outline

Unsupervised learning of sparse shift-invariant spatio-temporal features

Sequence classification with LSTM recurrent neural networks

Experimental results
Outline

Unsupervised learning of sparse shift-invariant spatio-temporal features

Sequence classification with LSTM recurrent neural networks

Experimental results
Proposed model

Spatio-Temporal Convolutional Sparse Auto-Encoder for Sequence Classification
M. Baccouche, F. Mamalet, C. Wolf, C. Garcia and A. Baskurt
The model operates on small **space-time patches** to reduce the diversity of the content to be encoded.
The model operates on small space-time patches to reduce the diversity of the content to be encoded.

\( Z_i \) : the learned features. \( W_E \) and \( W_D \) are the auto-encoder parameters.
The model operates on small space-time patches to reduce the diversity of the content to be encoded.

- $Z_i$: the learned features. $W_E$ and $W_D$ are the auto-encoder parameters.

- **Best shift search**: to handle spatio-temporal shift-invariance of the learned representation. *Details are given in the next slides*
The model operates on small **space-time patches** to reduce the diversity of the content to be encoded.

- $Z_i$: the learned features. $W_E$ and $W_D$ are the auto-encoder parameters.

- **Best shift search**: to handle spatio-temporal shift-invariance of the learned representation. *Details are given in the next slides.*

- **Sparsifying logistic**: a function which transforms a non sparse code to a sparse one. *Details are given in the next slides.*
Best shift search

- The spatio-temporal neighborhood of a given patch $X_i$ is represented with a single patch $\phi(X_i, t_i)$.

- The “best” shift is the one minimizing the objective function, given the current set of parameters — *details are given in the next slides*.

- $t_i$ is an additional hidden parameter of the objective function.
Sparsifying logistic

non sparse code

sparse code
Sparsifying logistic

- Sparsifying logistic: introduced by Ranzato et al.

\[ \bar{z}_i^{(k)} = \frac{\eta e^{\beta z_i^{(k)}}}{\xi_i^{(k)}} \quad \text{with} \quad \xi_i^{(k)} = \eta e^{\beta z_i^{(k)}} + (1 - \eta) \xi_{i-1}^{(k)} \]

- \( \eta \) controls the code sparsity. \( \beta \) controls the code softness.
- The sparsifying logistic introduces strong non-linearities: the encoder and the decoder must be trained separately.
Objective function

- $Z_i$ is considered as an additional parameter to train the model to produce the “optimal” code for the decoder.
- The hidden variables are: $(Z_i, t_i, W_E, W_D)$
Objective function

- $Z_i$ is considered as an additional parameter to train the model to produce the “optimal” code for the decoder.

- The hidden variables are: $(Z_i, t_i, W_E, W_D)$

- The objective function is the sum of two terms representing the encoder prediction and the decoder reconstruction errors:

$$E(X_i, t_i, Z_i, W_E, W_D) = E_E(X_i, t_i, Z_i, W_E) + E_D(X_i, t_i, Z_i, W_D)$$
$$= \|Z_i - Enc(W_E, \phi(X_i, t_i))\|^2 + \|Dec(W_D, \bar{Z}_i) - \phi(X_i, t_i)\|^2$$
Training procedure

- The parameters which minimize the objective function are:

\[
t_i^* = \arg \min_{t_i} E \left( t_i | X_i, Z_i, W_E, W_D \right) \tag{1}
\]

\[
Z_i^* = \arg \min_{Z_i} E \left( Z_i | X_i, t_i^*, W_E, W_D \right) \tag{2}
\]

\[
(W_E^*, W_D^*) = \arg \min_{W_E, W_D} E \left( W_E, W_D | X_i, t_i^*, Z_i^* \right) \tag{3}
\]

1. Random initialization of \((W_E, W_D)\)
2. For a given \(X_i\), perform an exhaustive search to find \(t_i^*\) which solves equation (1)
3. Given \(\phi(X_i, t_i^*)\), solve equation (2) by performing steepest descent on \(Z_i\) to find \(Z_i^*\)
4. Equation (3) is solved by updating \(W_E\) and \(W_D\) with standard back-propagation
The encoder contains N trainable 3D convolution kernels, each one having the same size than the input patch.

After the training, the sparsifying logistic turns to a classical logistic function (since $\zeta$ is replaced by a fixed value).

The decoder consist of a set of neurons fully connected to the sparse code layer.
Learned basis

- The output patch is a weighted sum of elementary patches called the **basis** of the representation space.
- The basis corresponds to the decoder responses when stimulated with sparse codes that have only one non-zero element equal to 1.
- In these examples, the patches size is 8x8x3.
- Note that no element of the basis is a shifted version of another one.

KTH dataset

GEMEP-FERA dataset
Outline

- Unsupervised learning of sparse shift-invariant spatio-temporal features
- Sequence classification with LSTM recurrent neural networks
- Experimental results
Hierarchical model:

- Each sequence is decomposed into space-time blocks.
- Each space-time block is decomposed into small space-time patches.

For each timestep, the feature vector corresponds to the concatenated responses of the small patches placed at the grid of possible locations in the space-time block.
Sequence classification

- Used classifier: bidirectional long short-term memory
  - A particular recurrent neural network introduced by Schmidhuber et al.
  - Adapted to long sequences processing (e.g. videos)
- Feature sequences are used to train a bidirectional RNN with LSTM neurons:
  - Feature sequence
  - Input layer
  - Hidden layer: 5 LSTM for each direction
  - Output layer

BMVC’12
Outline

- Unsupervised learning of sparse shift-invariant spatio-temporal features
- Sequence classification with LSTM recurrent neural networks
- Experimental results
KTH human actions dataset

- 25 persons performing 6 actions in 4 scenarios
- Two versions used in the literature:
  - KTH1: 599 long sequences (several iterations of the same action / video)
  - KTH2: 2391 short sequences (one iteration of the same action / video)
KTH: results

- Leave-one-out cross validation (avg. on 25 runs)
- Patches size: 8x8x3
- Code size: 192
- $\eta = 0.02$ and $\beta = 1.5$
- For comparison, we also report the results of the 2D model:

<table>
<thead>
<tr>
<th>Dataset</th>
<th>2D model</th>
<th>2D+t model</th>
</tr>
</thead>
<tbody>
<tr>
<td>KTH 1</td>
<td>93.70</td>
<td>95.83</td>
</tr>
<tr>
<td>KTH 2</td>
<td>90.76</td>
<td>93.74</td>
</tr>
</tbody>
</table>
## KTH: Comparison with related works

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Features</th>
<th>Method</th>
<th>Acc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>KTH 1</td>
<td>Learned</td>
<td>Ours</td>
<td>95.83</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Jhuang et al.</td>
<td>91.70</td>
</tr>
<tr>
<td></td>
<td>Hand-crafted</td>
<td>Gao et al.</td>
<td>96.33</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Chend and Hauptmann</td>
<td>95.83</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Liu and Shah</td>
<td>94.20</td>
</tr>
<tr>
<td>KTH 2</td>
<td>Learned</td>
<td>Ours</td>
<td>93.74</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ji et al.</td>
<td>90.20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Taylor et al.</td>
<td>90.00</td>
</tr>
<tr>
<td></td>
<td>Hand-crafted</td>
<td>Kim et al.</td>
<td>95.33</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ikizler et al.</td>
<td>94.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Gao et al.</td>
<td>92.45</td>
</tr>
</tbody>
</table>
GEMEP-FERA facial expressions dataset

- Initially presented for the FERA 2011 challenge
- 10 persons / 5 facial expressions
  - Train: 155 videos / 7 persons
  - Test: 134 videos / 6 persons (3 are present in the train)
Results

- Patches size: 8x8x3
- Code size: 128
- $\eta = 0.02$ and $\beta = 1.5$
- Score calculated by the challenge organizers
- Evaluation on both PI and PS configurations:

<table>
<thead>
<tr>
<th></th>
<th>Anger</th>
<th>Fear</th>
<th>Joy</th>
<th>Relief</th>
<th>Sadness</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>PI</td>
<td>100</td>
<td>93.33</td>
<td>95</td>
<td>68.75</td>
<td>46.67</td>
<td>80.75</td>
</tr>
<tr>
<td>PS</td>
<td>92.31</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>98.46</td>
</tr>
<tr>
<td>Overall</td>
<td>96.30</td>
<td>96</td>
<td>96.77</td>
<td>80.77</td>
<td>68</td>
<td>87.57</td>
</tr>
</tbody>
</table>
Comparison with related works

<table>
<thead>
<tr>
<th>Method</th>
<th>PI</th>
<th>PS</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours</td>
<td>80.75</td>
<td>98.46</td>
<td>87.57</td>
</tr>
<tr>
<td>Yang and Bhanu</td>
<td>75.23</td>
<td>96.18</td>
<td>83.78</td>
</tr>
<tr>
<td>Tariq et al.</td>
<td>65.50</td>
<td>100</td>
<td>79.80</td>
</tr>
<tr>
<td>Littlewort et al.</td>
<td>71.40</td>
<td>83.70</td>
<td>76.10</td>
</tr>
<tr>
<td>Dhall et al.</td>
<td>64.80</td>
<td>88.70</td>
<td>73.40</td>
</tr>
<tr>
<td>Meng et al.</td>
<td>60.90</td>
<td>83.70</td>
<td>70.30</td>
</tr>
<tr>
<td>Valstar et al.</td>
<td>44</td>
<td>73</td>
<td>56</td>
</tr>
</tbody>
</table>
Conclusions and future work

- A generic model for sequence classification, with fully automated feature construction process
- The best results on the GEMEP-FERA facial expressions dataset
- For the KTH human actions dataset:
  - The best results among learning-based methods
  - Among the very best even when compared to hand-crafted features methods
- Future work:
  - Address scale invariance
  - Application to other problems
  - Investigate the use of “non uniform” targets in the LSTM classification step, and handle the temporal localization of the hidden states with a CTC output layer
Thank you for your attention

This presentation is available online at: http://liris.cnrs.fr/moez.baccouche/Baccouche-talk_BMVC-12.pdf