ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

Sample efficient localization and stage prediction with autoencoders

Sebastian Hoch[1], Sascha Lange[2] and Janis Keuper[1]

1- Offenburg University - Institute for Machine Learning and Analytics - Germany

2- PSIORI GmbH - Germany

Abstract. Engineering, construction and operation of complex machines involves a wide range of complicated, simultaneous tasks, which potentially could be automated. In this work, we focus on perception tasks in such systems, investigating deep learning approaches for multi-task transfer learning with limited training data.
We show an approach that takes advantage of a technical systems' focus on selected objects and their properties. We create focused representations and simultaneously solve joint objectives in a system through multi-task learning with convolutional autoencoders. The focused representations are used as a starting point for the data-saving solution of the additional tasks. The efficiency of this approach is demonstrated using images and tasks of an autonomous circular crane with a grapple.

1 Introduction

The success of deep learning leads to broad use in industry and business. In our daily work, we use machine learning for the automation of complex technical systems, from single robots to full production lines. In this work, we will focus on a wood-yard crane that is used for autonomous filling a paper mill. To operate entire plants or single machines automatically, tasks such as the perception of the environment have to be performed. Usually, in industrial projects, the annotations have to be generated newly for each task. They, therefore, contribute significantly to the project costs. In addition to annotation costs, rare events make it challenging to generate an appropriate data set. In our environment, for example, new logs are delivered by truck ten times a day. It is precisely these rare events that are of particular interest for the automation of technical systems. Technical systems are limited by the lack of data caused by the cost of labeling and the challenges of covering all special cases. The focus of the tasks on one object motivates us to use this information when solving the tasks. We contend that the need for samples in solving tasks can be reduced by domain knowledge, in our case the focus on the grapple. For this purpose, we have combined finding a suitable representation with multi-task learning (MTL) [1] in the version hard parameter sharing.

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

2 Stage prediction

Multi-task learning aims to pass on knowledge by learning some related tasks simultaneously. In [2], the authors follow this pattern without deviations. Reference [3] supplements the model by cascading one task's output as input to the next task. The paper [4] aims to encode the network structure in a continuous low-dimensional embedding space. Our approach is closest to the standard shared trunk approach, with elements of [3] and [4].
We create representations with the help of the autoencoder approach [5]. An autoencoder first compiles the input data into a low-dimensional space and then reconstructs it from this representation. No labels are required; the process is unsupervised. In the autoencoder environment, layer-wise pre-training is established, as introduced by [6]. This approach became popular through the work of [7] and [8].
In layer-wise pre-training, individual layers are trained with each other from the outside to the inside. This leads to a good initialization of the weights of the individual layers.
In our work, a layer-wise pre-trained autoencoder is combined with a classifier or regressor. These models' focus is on embedding spaces that allow solving more tasks with a small amount of data. A transfer within a domain from one label to another should take place. Our method is described in Fig 1.

3 Experiments and Results

We performed the experiments on a dataset which contains 5700 images, taken from a circular crane pointing towards the grapple. The grapple is visible in each image and has two different annotations. Due to the angle of the sun and weather influences, the brightness fluctuates significantly. The first annotation describes if the grapple is loaded. It is called 'grapple loaded?', is a classification task and is evaluated in our experiments with the metric accuracy. The second annotation describes the position of the yellow tip in the image using the x and y coordinates. We define a prediction for the keypoint yellow tip as correct if it is within a radius of eight pixels of the truth. To get a clear score, the number of correct predictions is divided by the number of all predictions (recall).

3.1 Experiments

In order to be able to evaluate our approach, we compare our results with a representation-based approach and a standard deep neural network (ResNet50 [9]). Each experiment is performed eight times, the average and standard deviation are presented as a result. For the evaluation, 10% of the data are used as test data.
Since all images in our dataset are annotated for both tasks, it makes sense to try our method in both directions. For this purpose, a focused autoencoder is trained for each label. Afterward, the found encoder with its representation is used for the transfer learning stage of the other label. The second stage of our

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

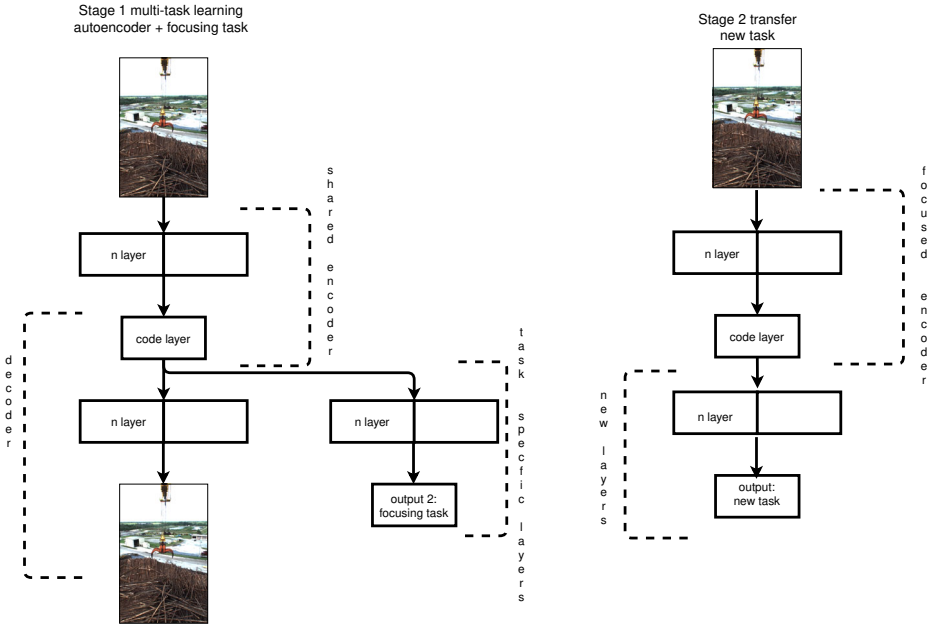

Fig. 1: Approach: **Stage 1**: A layer-wise pre-trained autoencoder is extended with a focusing task. The loss function, especially for the shared encoder, consists of the weighted sum of the two individual losses. **Stage 2**: The encoder (architecture + weights) of the multi-task learning model is extended with layers for solving one other task in the domain.

method is trained with all data and with few (1500) data. Particular interest is given to the performance that can be achieved with little data compared to the other approaches.
The multi-task models are designed according to Section 2. As input, the images are resized from the size (1024,648,3) to the size (256,192,3). The Encoder network consists of 12 convolution layers along with three fully connected layers. The kernel size is always (3,3). Subsequent to the input layer, the first three convolution layers have eight filters followed by two layers of 16 filters, three layers of 32 filters and four layers with 64 filters. The hyperparameter stride is set to two in layers 3, 5, 8, 10 and 12. In the other layers, the hyperparameter strides is set to one. The activation of layers 3, 5, 8 and 10 is normalized by batch normalization [10]. The fully connected layers have the sizes 512, 128 and 32. The code layer has seven neurons and is activated linearly. The decoder is designed symmetrically to the encoder. Thereby convolutional layers have been replaced by a transposed convolutional layer.
The focus criterion model used for the classification 'grapple loaded?' consists of three fully connected layers with 24, 8, and 1 neuron. Apart from the output layer, the activation's are adjusted using batch normalization. The key point

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

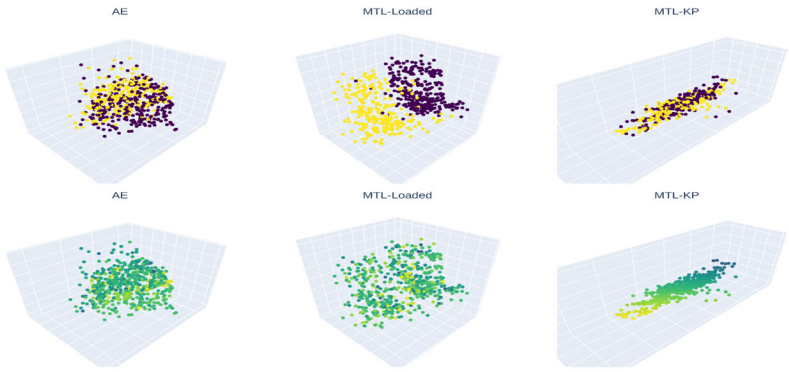

Fig. 2: Embeddings of the different models. The colored markings in the top row correspond to the annotation 'grapple loaded?' and in the bottom row to the grapple's y position. The MTL models show strong adaptation to their focusing task, while the normal autoencoder shows no adaptation to the tasks.

detection yellow tip model differs from this model by having 32 neurons and linear activation in the first layer.
The networks for the second stage consist of the trained and focused encoder. To solve the actual task, the encoder is extended by the task-specific layers. The same architecture as just described for the focusing part is used.
For comparison purposes, the tasks are learned using standard autoencoders and subsequent transfer learning. In the results tables these results are simply titled as autoencoder.

3.2 Embedding

To be able to evaluate an embedding, the instrument of visualization is applied. For this purpose, the test data is encoded and provided with the corresponding annotation. For ease of presentation, a principal component analysis is performed on the coded data for three components. Fig. 2 shows the embeddings of the three models marked with the two annotations. The autoencoder model, like all other models, manages to encode the data points. No adaptation is made to the grapple in the autoencoder model, nevertheless. The MTL models show a clear adaptation to their focusing task. Since we emphasize the grapple, we rate the encoding of the position higher than the separation by class.

3.3 Classification 'grapple loaded?'

The classification task 'grapple loaded?' is well solved by all approaches. Table 1 lists the results for 5000 and 1500 data points for the different methods. With

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

Table 1: Model performance of the selected models (%), which are trained by different data volume, for both tasks. The best result is highlighted in bold. For a few data, the MTL and Transfer approach shows significantly better performance than the other approaches.

data	ResNet50	Autoencoder	STAGE 1 MTL	STAGE 2 Transfer
key point detection: yellow tip				
5000	**0.895 ±0.09**	0.865 ±0.04	0.862 ±0.032	0.884 ±0.027
1500	0.19 ±0.178	0.522 ±0.153	-	**0.568 ±0.212**
classification: grapple loaded?				
5000	0.931 ±0.02	0.902 ±0.013	0.901 ±0.025	**0.933 ±0.01**
1500	0.768 ±0.04	0.709 ±0.098	-	**0.878 ±0.026**

5000 images, all approaches are over 90% correct. The model which was pretrained with the keypoint yellow tip is the best. It achieves an accuracy of 93.3%, closely followed by the ResNet50 with 93.1%. At 1500 images, our approach achieves an accuracy of 87.8 %. This is significantly better than the second-placed ResNet50 with a performance of 76.6%.

3.4 Keypoint detection 'yellow tip'

An examination of the results for the keypoint detection task reveals a similar picture to the classification task. The presented method is best, especially when there is little data. The experimental results are shown in table 1. At 1500 data points, the representation-based approaches are significantly better than the ResNet50. Whereby the presented approach outperforms the autoencoder approach by 4.6%. At 5000 data points, the ResNet50 delivers the best performance, closely followed by our approach. As with the other task, the MTL part of our approach delivers similar performance to the autoencoder-based approach. When all the results are considered together, it can be seen that the method presented delivers better results than the other approaches, especially when only a small amount of data is available. If more data is available, the approach is competitive but does not always produce the best performance. The comparison with a convolutional autoencoder shows that a suitable focus of the embedding can be advantageous for subsequent tasks. It is also evident that different tasks solve this differently well. A task that shows a clearer relation to the relevant region seems to be more suitable. The focusing task neither loses nor gains from the MTL but is essential to the focus of the representation. Based on the performance shown, the approach is particularly recommended when several tasks with a common focus need to be solved. Costs can be saved due to the reduced need for data. The standard neural network shows its strength with many data and through its lower training effort.

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

4 Conclusions

This paper has presented our first approach towards focused representations for the solution of multi-task recognition problems in a data-efficient manner. The approach modifies autoencoder by a shared encoder and a focusing task. Simultaneous training of the autoencoder and arbitrary related tasks allows to focus the autoencoders prediction capabilities onto relevant regions in images. These representations are used to solve further tasks of a technical system in a data-saving way.
To demonstrate this method's performance, the approach was compared with a standard deep neural network and a representation-based approach on an industry dataset. Compared to a classical autoencoder, the found representations are focused on regions of interest.
We recommend that future work be directed in particular to improving representation search. For this purpose, the effect of additional parallel tasks can be investigated and different strategies for the loss function can be analyzed.
The performance of the proposed approach was demonstrated using a real-world dataset. We believe that such data-efficient approaches can significantly reduce annotation costs, especially in industrial projects.

References

[1] Rich Caruana. Multitask learning. In Sebastian Thrun and Lorien Pratt, editors, *Learning to Learn*, pages 95–133. Springer US, Boston, MA, 1998.

[2] Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Facial landmark detection by deep multi-task learning. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 94–108, Cham, 2014. Springer International Publishing.

[3] Jifeng Dai, Kaiming He, and Jian Sun. Instance-aware semantic segmentation via multi-task network cascades, 2015.

[4] Shanfeng Wang, Qixiang Wang, and Maoguo Gong. Multi-task learning based network embedding. *Frontiers in Neuroscience*, 13, 01 2020.

[5] D. E. Rumelhart and J. L. McClelland. Learning internal representations by error propagation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations*, pages 318–362. 1987.

[6] D. Ballard. Modular learning in neural networks. In *AAAI*, 1987.

[7] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science (New York, N.Y.)*, 313:504–507, 2006.

[8] Y. Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy layer-wise training of deep networks. *Adv. Neural Inf. Process. Syst.*, 19:153–160, 01 2007.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[10] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015.