In: Proc. of the 25th Int. Conference on Artificial Neural Networks (ICANN)
Part II, LNCS 9887, pp. 80-87, Barcelona, Spain, September 2016
The final publication is available at Springer via
http://dx.doi.org//10.1007/978-3-319-44781-0_10

The Effects of Regularization on Learning Facial Expressions with Convolutional Neural Networks

Tobias Hinz, Pablo Barros, and Stefan Wermter

University of Hamburg Department of Computer Science,
Vogt-Koelln-Strasse 30, 22527 Hamburg, Germany
{4hinz,barros,wermter}@informatik.uni-hamburg.de
http://www.informatik.uni-hamburg.de/WTM

Abstract. Convolutional neural networks (CNNs) have become effective instruments in facial expression recognition. Very good results can be achieved with deep CNNs possessing many layers and providing a good internal representation of the learned data. Due to the potentially high complexity of CNNs on the other hand they are prone to overfitting and as a result, regularization techniques are needed to improve the performance and minimize overfitting. However, it is not yet clear how these regularization techniques affect the learned representation of faces. In this paper we examine the effects of novel regularization techniques on the training and performance of CNNs and their learned features. We train a CNN using dropout, max pooling dropout, batch normalization and different combinations of these three. We show that a combination of these methods can have a big impact on the performance of a CNN, almost halving its validation error. A visualization technique is applied to the CNNs to highlight their activations for different inputs, illustrating a significant difference between a standard CNN and a regularized CNN.

Keywords: convolutional neural network, facial expression recognition, regularization, batch normalization, dropout, max pooling dropout

1 Introduction

The increasing size and complexity of neural networks in the recent past give more freedom to developers and provide solutions for more complex problems, but also make them more prone to overfit the given input data. This is especially the case in supervised settings when there is only a very limited amount of training data.

To deal with this problem various regularization methods have been developed to reduce overfitting. These techniques include established techniques such as early stopping, where training is stopped as soon as the validation error stops to improve and L2 regularization, the neural network equivalent of the Ridge regression. More recently new methods for regularization were introduced, such as dropout [2], drop-connect [13], max pooling dropout [3], stochastic pooling [4] and to some degree batch normalization [5].

Dropout, max pooling dropout and batch normalization have been introduced in the previous four years. While they have been used and examined individually, the authors know of no work in which all three methods are tested and evaluated in conjunction with each other.

This research applies some of the most recently developed regularization methods to a CNN trained on images from the Cohn-Kanade dataset [7]. The Cohn-Kanade dataset contains human faces expressing different emotions, such as happiness, anger or surprise. We train a CNN on this dataset, using dropout, max pooling dropout and batch normalization. The effect of different combinations of these three techniques on the training of the CNN is examined by monitoring the development of the validation error over time, as well as by visualizing CNNs' activations for different input images.

2 Background

In this chapter we will first give a brief overview over the tested regularization methods, i.e. dropout [2], max pooling dropout [3] and batch normalization [5].

2.1 Dropout

In 2012 Hinton et al. [2] introduced the dropout method to prevent artificial neural networks from overfitting. Dropout prevents co-adaptation of the network's weights to the training data. To achieve this each hidden unit of the network is omitted with a given probability - usually 0.5 - for any training sample.

This means that for each training sample a selected subset of units, including their incoming and outgoing connections, are temporarily removed from the network. If a dropout probability p of 0.5 is used, roughly half of the activations in each layer are deleted for every training sample, thus preventing hidden units from relying on other hidden units being present.

For testing the network on independent test data, the "mean network" is used. It contains all the hidden units, but has to compensate for the fact that during testing roughly twice as many hidden units are active, compared to the training phase. Due to this the weights are rescaled proportional to the dropout probability, for example for a dropout probability of 0.5 all weights are divided by two [2].

2.2 Max Pooling Dropout

Max pooling dropout is a dropout variant especially designed for CNNs, introduced by Wu and Gu [3]. In a standard CNN we have alternating convolutional and pooling layers. Common pooling mechanisms include for example max or average pooling. Wu and Gu suggested using dropout within the pooling layers to introduce stochasticity into the training process. Instead of deterministically choosing the strongest activation in the pooling region, max pooling dropout allows smaller activations to be chosen instead.

To achieve this, dropout is applied to each pooling regions, before max pooling is performed. Using max pooling dropout is therefore sampling from a multinomial distribution to select an index i to choose the pooled activation a_i. As such max pooling dropout can be seen as a special variant of stochastic pooling [4], with the difference that activations are used with a probability proportional to their rank, instead of the strength of their activation.

2.3 Batch Normalization

During training the distribution of inputs to a given layer changes as parameters in the previous layer are updated. Therefore, parameter initialization and the learning rate can have a high impact on the progress of the training. This phenomenon, also called internal covariate shift, is addressed by the technique called batch normalization [5]. Batch normalization works by normalizing each layer's input for each mini batch during training. This allows much higher learning rates, more freedom regarding parameter initialization and also acts as a regularizer.

To that end each layer's input is normalized. To preserve what each layer can represent for each activation $x^{(k)}$, a pair of parameters $< \gamma^{(k)}, \beta^{(k)} >$ is introduced, which scales and shifts the normalized values. These additional parameters are learned along with the original model parameters and make sure the representational capability of the network is not changed.

Batch normalization can work as a form of regularization, since a training example is seen in conjunction with other examples in a mini batch. Due to shuffling, the composition of mini batches changes during training, so the network no longer produces deterministic values for a given training example.

3 Methodology

To test the previously described regularization methods we examined a CNN and trained it to classify images from the Cohn-Kanade dataset [7]. The Cohn-Kanade dataset consists of images depicting human faces in seven emotions: anger, contempt, disgust, fear, happiness, sadness and surprise. In line with other research [8] we only used six classes, neglecting contempt for our training and testing. Each example of emotion contains a sequence of up to 60 frames, that starts with a neutral expression and continues to the peak of the expression. Our training and testing set comprised the last three images of each sequence. These images were rescaled to 128 × 128 pixels, converted to gray scale and whitened.

3.1 Experiments

The CNN used for the experiments consists of six layers. The first three layers are convolutional layers, followed by two fully connected layers and one softmax layer for classification on top. Max pooling is performed after each convolutional layer

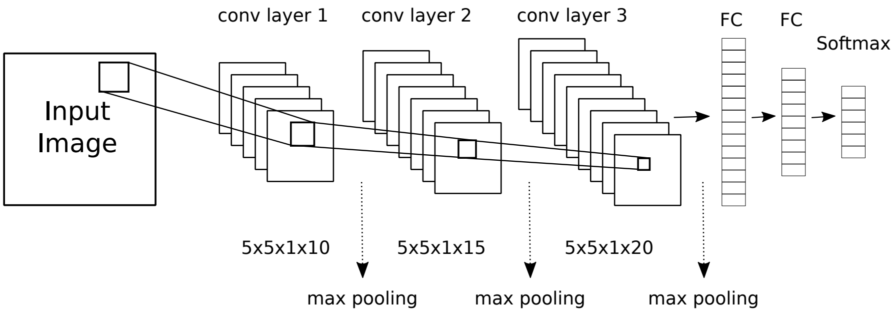

Fig. 1. CNN architecture: our CNN consists of three convolutional layers with 10, 15 and 20 filters, each with a filter size of 5×5. 2×2 max pooling is performed after each convolutional layer. The convolutional layers are followed by two fully connected layers with 500 and 200 units and one classification layer with 6 units.

and the number of filters per convolutional layer are 10, 15 and 20 respectively. A filter size of 5×5 is used on each convolutional layer. The two fully connected layers consist of 500 and 200 units and the logistic regression layer has 6 units for classification, see Figure 1.

As activation function ReLU was used on all layers and weight initialization was performed according to current guidelines [6]. The initial learning rate is 0.001, which is linearly reduced by 1% per epoch. A momentum of 0.9 was used and L2 regularization with a small penalty of 0.0001 was introduced since it improved stability during training.

With this fixed architecture we then proceeded to test the effects of the different methods on the set classification task. The following eight settings were tested:

1. no regularization,
2. standard dropout after each layer,
3. max pooling dropout after each convolutional layer,
4. batch normalization (BN) after each layer,
5. max pooling dropout after each convolutional layer and standard dropout after each fully connected layer,
6. max pooling dropout after each convolutional layer and BN after each layer,
7. standard dropout after each layer and BN after each layer,
8. max pooling dropout after each convolutional layer and standard dropout after each fully connected layer and BN after each layer.

For each individual setting training was performed using stochastic gradient descent for a total of 150 epochs. We split our dataset into ten independent subsets of equal size and performed 10-fold cross-validation in the manner presented by Liu et al. [9]. After training was completed we applied a visualization technique [10] to our CNN, to demonstrate the potential impact of regularization methods on the learned features. For this we deconvolve our CNN and then visualize the activations of the third convolutional layer for various input images.

3.2 Results

The most important evaluation criterion for the proposed methods is whether they are able to decrease the validation error, i.e. improve the system's generalization capability. Figure 2 depicts plots of the development of the validation error over time for each regularization method. The plots show the average validation error of all runs for a given regularization method and the combination of that method with batch normalization. Table 1 gives the average best validation error and the standard deviation of the best validation errors for each tested regularization method.

Except for batch normalization each combination of regularization methods outperformed no regularization. The improvements from all combinations from max dropout and batch normalization onward are statistically significant when compared to no regularization. The results also indicate that the combination of several regularization methods, as opposed to using one single method, further improves regularization. Each combination of at least two regularization methods performed better than using only one single method.

Table 1. Average accuracy and standard deviations for 10-fold cross-validation for the combinations of different methods. Sorted in order of increasing accuracy.

Method	Accuracy	Std
Batch Normalization (BN)	86.9%	4.4
No Regularization	89.6%	4.5
Dropout	92,4%	3.3
Max Dropout	92,8%	3.4
Max Dropout + BN	93.3%	4.0
Dropout + BN	93,9%	4.1
Max Dropout + Dropout + BN	94.3%	4.2
Max Dropout + Dropout	94.3%	2.5

The addition of batch normalization to any regularization methods did not improve the final accuracy. However Figure 2 shows quite clearly that the addition of batch normalization had the advantage of converging quicker to better results. Since the differences in the results between any regularization method and that regularization method in combination with batch normalization are not statistically significant, it seems that the addition of batch normalization helps the training process.

Figure 3 shows a visualization [10] of each filter on the third convolutional layer for the input image depicted in the respective leftmost column. The images on the left of the second and third column depict the activations of a standard CNN trained without regularization and the images on the right a regularized CNN trained with the combination of max dropout and dropout. It can be seen that the activations of the regularized CNN are much more focused on certain parts of the face, while the standard CNN is activated for much bigger regions. This can explain the higher accuracy of the regularized CNN, as the regularization methods seem to force it to focus on certain aspects of the face. The filters of the standard CNN on the other hand are often quite blurry and indistinct, explaining its lower accuracy.

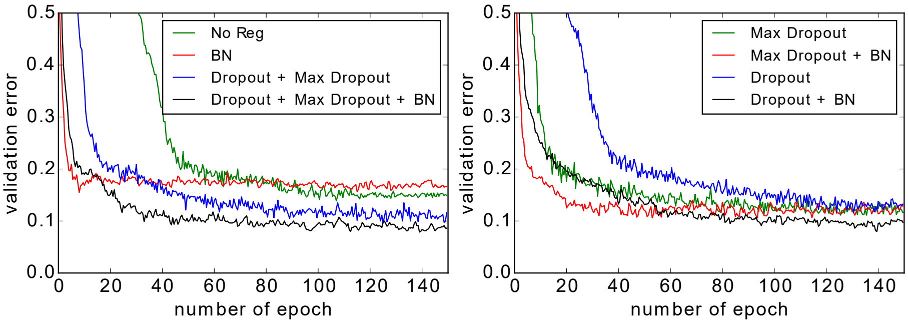

Fig. 2. Development of validation errors during training time.

3.3 Discussion

The differences in the filters' activations shown in Figure 3 for the third convolutional layer between the standard and the regularized CNN are notable. Many of the standard CNN's filters do not focus on specific parts of the face, but are instead spread over the whole input. As a result we have many activations in areas of the input that are not relevant to the classification, such as the corners of the image.

In all images of the Cohn-Kanade dataset the faces are quite centered in the image and as a result the corners of the inputs do not provide relevant information for the classification task. This is reflected by the activations of the regularized CNN, which are mostly focused on the facial features themselves. Here the filters are much more selective and mostly focus on the center part of the image. This focus is likely to improve the overall accuracy of the CNN compared to one without applied regularization.

Indeed, Khorrami et al. [8] showed in their work that the most important features are centered around the eyes, the nose and the mouth. The visualizations show that the regularized CNN mainly focuses on these areas. It is also noteworthy that our accuracy is comparable to previous results [8],[11], [12]. While we do not achieve state-of-the art accuracy it has to be noted that we do not perform data augmentation and only use roughly a tenth of the number of filters as e.g. Khorrami et al. [8]. It can be expected that the accuracy of our network can be further improved by utilizing data augmentation techniques even without increasing the number of used filters.

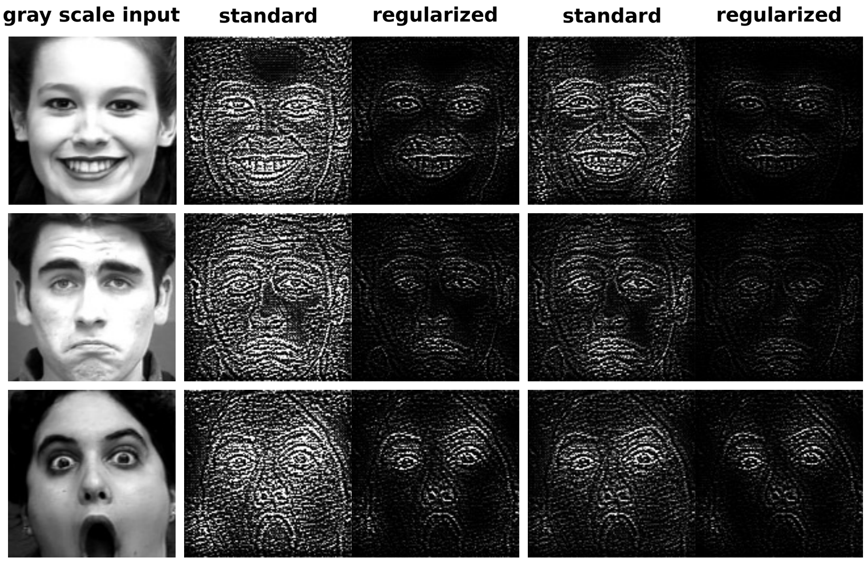

Fig. 3. Visualization of third convolutional layer's activations for the input image on the left of each row. The left image of the second and third column depicts the activations of a standard CNN, while the right image of the respective column shows the activations of the same filter in a CNN regularized with a combination of max pooling dropout and common dropout.

4 Conclusion

In this work we showed that for the training of a CNN the combination of max dropout and standard dropout can achieve very high accuracy on the Cohn-Kanade dataset, even without applying data augmentation and with a comparatively small number of used filters. A visualization of the trained networks shows a big difference between a regularized and a standard CNN, exemplifying the effects of regularization firsthand. While the standard CNN's filters are often blurry and indistinct, the regularized CNN's filters exhibit a much higher selectivity and are more focused on important features.

In our experiments batch normalization had no effect on the generalization capability of a trained CNN. However, it did not affect the accuracy of a CNN in a negative way, while simultaneously reducing the training time until good results are achieved. It therefore seems that the addition of batch normalization to the training procedure is advantageous.

Finally, we have shown that with the right combination of applied regularization techniques it is possible to achieve good results with small networks and without data augmentation. In the future, these regularization techniques can be applied together with data augmentation and more complex CNNs, either with more filters or more layers, to potentially achieve an even higher accuracy on challenging datasets.

Acknowledgments This work was partially supported by the CAPES Brazilian Federal Agency for the Support and Evaluation of Graduate Education (p.n.5951–13–5), the German Research Foundation DFG under project CML (TRR 169), and the Hamburg Landesforschungsförderungsprojekt.

References

1. Krizhevsky, A., Sutskever, I., Hinton, G. E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp. 1097–1105 (2012)
2. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research 15(1), 1929–1958 (2014)
3. Wu, H., Gu, X.: Towards dropout training for convolutional neural networks. Neural Networks 71, 1–10 (2015)
4. Zeiler, M. D., Fergus, R.: Stochastic pooling for regularization of deep convolutional neural networks. arXiv preprint 1301.3557 (2013)
5. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint 1502.03167 (2015)
6. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1026–1034. (2015)
7. Lucey, P., Cohn, J. F., Kanade, T., Saragih, J., Ambadar, Z., Matthews, I.: The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In: Proceedings of the Third International Workshop on CVPR for Human Communicative Behavior Analysis, pp. 94–101. (2010)
8. Khorrami, P., Paine, T., Huang, T.: Do Deep Neural Networks Learn Facial Action Units When Doing Expression Recognition?. In: Proceedings of the IEEE International Conference on Computer Vision Workshops, pp. 19–27. (2015)
9. Liu, M., Li, S., Shan, S., Chen, X.: Au-aware deep networks for facial expression recognition. In: 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG), pp. 1–6. (2013)
10. Zeiler, M. D., Fergus, R.: Visualizing and understanding convolutional networks. In: Computer visionECCV, pp. 818–833. Springer International Publishing (2014)
11. Barros, P., Weber, C., Wermter, S.: Emotional expression recognition with a cross-channel convolutional neural network for human-robot interaction. In: IEEE-RAS 15th International Conference on Humanoid Robots, pp. 582–587. (2015)
12. Liu, P., Han, S., Meng, Z., Tong, Y.: Facial expression recognition via a boosted deep belief network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1805–1812. (2014)
13. Wan, L., Zeiler, M., Zhang, S., Cun, Y. L., Fergus, R.: Regularization of neural networks using dropconnect. In: Proceedings of the 30th International Conference on Machine Learning (ICML-13), pp. 1058–1066. (2013)