IEA: INNER ENSEMBLE AVERAGE WITHIN A CONVOLUTIONAL NEURAL NETWORK

Anonymous authors
Paper under double-blind review

ABSTRACT

Ensemble learning is a method of combining multiple trained models to improve the model accuracy. We introduce the usage of such methods, specifically ensemble average inside Convolutional Neural Networks (CNNs) architectures. By Inner Average Ensemble (IEA) of multiple convolutional neural layers (CNLs) replacing the single CNL inside the CNN architecture, the accuracy of the CNN increased. A visual and a similarity score analysis of the features generated from IEA explains why it boosts the model performance. Empirical results using different benchmarking datasets and well-known deep model architectures shows that IEA outperforms the ordinary CNL used in CNNs.

1 INTRODUCTION

Ensemble learning Zhou (2012); Rokach (2010) is the method of combining multiple models trained over the same dataset or random sets of the dataset like the methods of bagging and boosting Opitz & Maclin (1999) to improve the model prediction. A different variation is to ensemble the same model at different training epochs Qiu et al. (2014). The methods of ensemble have been widely used in deep learning Qiu et al. (2014); Dietterich (2000); Drucker et al. (1994) to improve the overall model accuracy. Ensembling neural networks is known to reduce the variance in prediction, in other words it helps the network to generalize more than the usage of a one network see for instance Krogh & Vedelsby (1995); Geman et al. (1992); Zhou et al. (2002). The work done by Stahlberg & Byrne (2017) proposed a method of ensembling multiple models where it unfolds the ensemble into a larger network. Lee et al. (2015) discussed the power of ensemble in training CNNs, they proposed a method for training ensemble by a specific loss function rather than averaging the predictions of the models.

Convolutional neural networks (CNNs) Lecun et al. (1998) are extremely successful architectures that are widely used in different areas like computer vision Krizhevsky et al. (2012), text analysis dos Santos & Gatti (2014); Lai et al. (2015) and even in general temporal sequence problems Bai et al. (2018). CNNs are a biologically inspired simulation of the cats visual cortex Hubel & Wiesel (1968). Usually CNNs are composed of a convolutional neural layer (CNL) followed by a pooling layer. The CNL and pool layers are repeated and stacked to introduce more depth inside the model being constructed. We call the connection of CNLs and pool layers a features extraction head. The features extraction head may or may not be connected to a fully connected layer (FCL) based on the application of the deep model. One important feature of CNLs that the weights are shared for creating the features. By having this feature of shared weights, CNLs does not strongly contribute to the total parameter size of the deep model unlike the contribution of the FCL.

In this work we show empirically, visually and trough a similarity scores analysis that replacing ordinary CNL by an Inner Ensemble average (IEA) of CNLs in a CNN can reduce the variance of this model. To the best of our knowledge we are the first to introduce the usage of ensemble average within a CNN architecture instead of combining multiple trained CNNs.

This work is organized as fellow, section 2 defines IEA mathematically and discusses the suitable number of ensembles needed. Section 3 shows the experiments performed on different benchmark datasets like MNIST dataset LeCun (1998), rotated-MNIST dataset Larochelle et al. (2007) and CIFAR-10 dataset Krizhevsky & Hinton (2009) using well-known deep CNNs architectures. We show the results of CNL only models, IEA of CNLs models and ensemble of both techniques. In

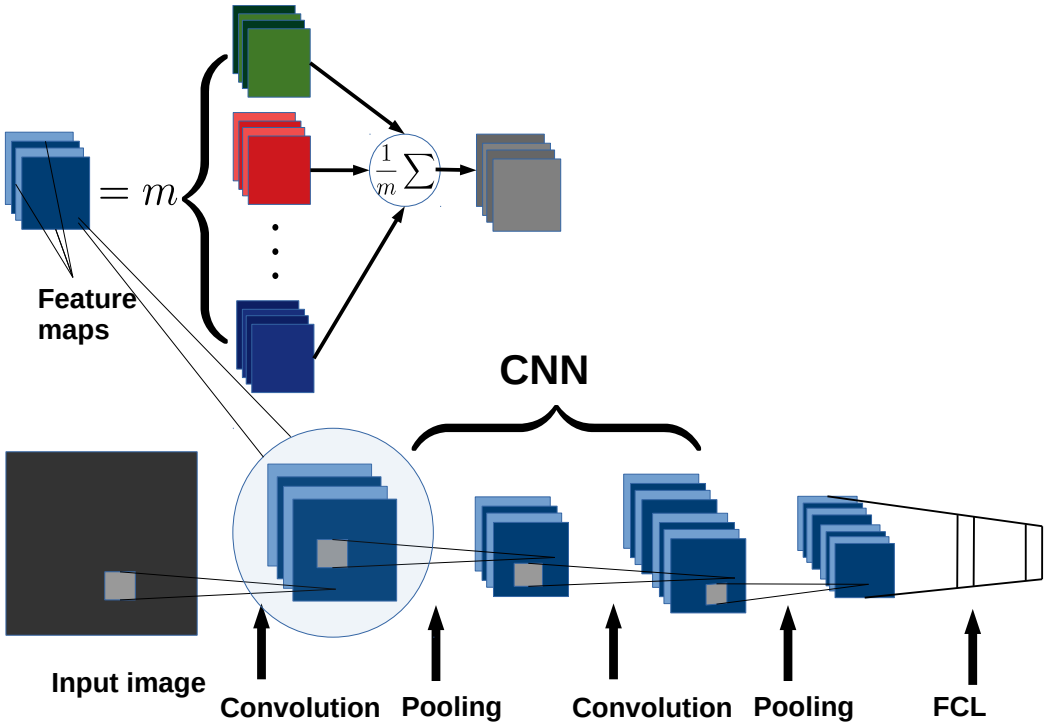

Figure 1: An illustration of the IEA concept. IEA modifies a CNN by replacing each CNL with an IEA layer.

sections 4 a visual analysis of the features generated from IEA usage is performed. Also, in section 4 we introduce a metric to measure the similarity between the features generated from IEA and CNLs.

2 Inner Ensemble Average (IEA)

2.1 IEA Definition

The IEA concept can be applied to a CNN architecture by replacing any CNL (C_{layer}) with average ensemble of CNLs. In any deep CNN architecture to use the IEA concept the C_{layer} is replaced by C_{IEA}. The C_{IEA} is defined as follows:

$$C_{IEA} = \frac{1}{m}(\sum_{i=1}^{m} C_{layer=i}) \tag{1}$$

where $m = \{x \,|\, x \in \mathbb{N}^{+} - \{1\}\}$ is the number of inner CNLs which is a design choice. An illustration of IEA concept within a CNN is found in figure 2.

When using IEA, the same settings of the replaced CNL are applied to each IEA element individually.

2.2 Influence of the number of ensembles m

We executed a set of tests using one, two and three layers deep CNN models that were trained over the rotated-MNIST dataset to find the right number of inner ensembles m. The configuration of training settings is mentioned in section 3.1. Each CNL is replaced by an IEA of CNLs. The m in each IEA layer has been set to integers values ranging from 1 to 30. A total number of 90 models were trained. Figure 2 shows the influence of m on the accuracy of the CNNs. The accuracy of the CNN increases for low values of m, and saturates or decreases at some point (in the present case when $m \gtrapprox 7$).

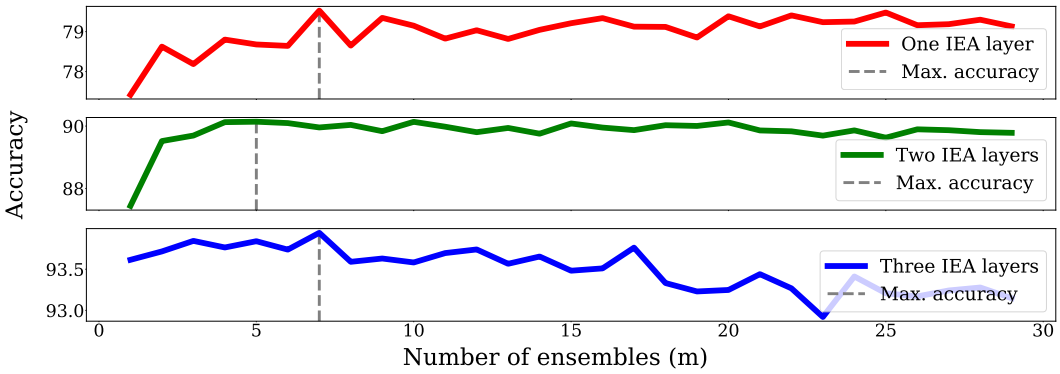

Figure 2: The influence of changing the number of inner ensembles m on model accuracy.

Note that this threshold could be a function of several model hyper-parameters (including kernel size, number of channels....).

3 Experiments

In the following section, we empirically evaluate the performance of IEA CNLS versus ordinary CNL. In all tests, the number of inner ensembles is set to $m = 3$, to speed up computations. We also compare the results of average of an ensemble of models trained using CNL and an ensemble of model trained using IEA of CNLs. The ensemble of IEA of CNLs can be described as an outer and inner ensemble in a CNN.

3.1 Performance analysis on the MNIST and rotated-MNIST datasets

The MNIST dataset is one of the earliest datasets used in computer vision and deep learning for evaluating different models performance. The MNIST dataset contains 60,000 training set examples and a test set of 10,000 examples of hand written digits. The rotated-MNIST dataset contains 62000 randomly rotated handwritten digits. The rotated-MNIST dataset is split into a training and test sets of size 50000 and 12000 respectively. Several models were trained on both datasets. We varied the number of layers in which we control how deep these models are. Each model contains several layers. Each layer is composed of a convolution layer batch normalized Ioffe & Szegedy (2015) and followed by a rectified linear unit (ReLU) Hahnloser et al. (2000) activation function. Then a max pool layer is stacked. After stacking the desired number of layers an average pool layer is used before the connection to a classification head. The models were trained using stochastic gradient descent (SGD) Bottou (2010) optimization algorithm with a momentum Polyak (1964) set to 0.9 and weight decay Krogh & Hertz (1992) set to 5e-4. The initial learning rate is 0.1 and it was divided by 10 for every 100 training epochs. The total number of training epochs is 350. The training datasets were not augmented, and the same initialization settings were kept the same between the models.

Table 1 describes the results on the MNIST test dataset. The usage of IEA of CNLs leads to an increase in performance over CNL only models. It can be seen in the case of one-layer deep model that the error rate of the model is decreased by 0.18%. By going deeper with one more layer, the mean error rate improved by 0.76% with just two IEA of CNLs. We achieve a mean error rate of 0.45% on the MNIST dataset by just using three layers deep model. Table 2 shows the average ensemble of three IEA of CNLs model performance versus an ensemble of three CNL only models. In the case of one-layer deep model, both ensemble of IEA of CNLs and CNL only model show the same performance. The ensemble of IEA models shows a better performance than the ensemble of CNL only based model when two or three layers are used.

Table 3 describes the rotated-MNIST test dataset results. In this case,the usage of IEA of CNLs still leads to significant improvements over CNL only models. It can be seen in the case of one-layer model that the IEA of CNLs model significantly improve the accuracy compared to the CNL only

model. The performance increases when more layers are added (deeper model). We can see a drop of the mean error rate by 0.41% in the case of two layers deep model, and a drop of 0.68% with three layers deep one. These results show the capability of IEA in decreasing the total model variance. Table 4 shows the case of average ensemble of three IEA of CNL models and CNL only based models. In the case of a one layer deep model, the ensemble of CNL models is better than the ensemble of IEA of CNL models. When more layers are considered, the ensemble of IEA of CNL models outperforms the ensemble of CNL only models, except in the case of two layers model were the ensemble of CNL only models has a slightly better performance. We due this behavior to the selection of hyper-parameters. Also, an error rate of 5.33% is achieved using three layers deep model with IEA of CNLs.

Table 1: Test set mean error rates in% with standard deviation on the MNIST dataset with different configurations of the tested model architecture.

Model depth	CNL	IEA (ours)
1-layer	1.52± 0.02	**1.34± 0.03**
2-layers	1.45± 0.03	**0.69± 0.01**
3-layers	1.46± 0.12	**0.45± 0.02**

Table 2: Test set average ensemble error rates in% on the MNIST dataset with different configurations of the tested model architecture.

Model depth	Ensemble of models using CNL	Ensemble of models using IEA (ours)
1-layer	1.32	1.32
2-layers	1.39	**0.68**
3-layers	1.48	**0.38**

Table 3: Test set mean error rates in % with standard deviation for the rotated-MNIST dataset with different configurations of the tested model architecture.

Model depth	CNL	IEA (ours)
1-layer	47.06± 0.33	**21.18± 0.37**
2-layers	10.47± 0.09	**10.06± 0.05**
3-layers	6.72± 0.27	**6.04± 0.04**

Table 4: Test set average ensemble error rates in % for the rotated-MNIST dataset with different configurations of the tested model architecture.

Model depth	Ensemble of models using CNL	Ensemble of models using IEA (ours)
1-layer	45.61	**19.27**
2-layers	**8.93**	8.95
3-layers	5.86	**5.33**

3.2 Performance analysis on the CIFAR-10 dataset

The CIFAR-10 dataset consists of 60000 color images of size 32x32 and contains 10 classes. There are 50000 training images and 10000 test images. A well-known object detection models like VGG16 Simonyan & Zisserman (2014), residual network with 18 layers (ResNet18) and 101 layers (RestNet101) He et al. (2015), Mobilenet Howard et al. (2017) and Densely Connected Convolutional Networks with 121 layers (DenseNet) Huang et al. (2016) were trained on the CIFAR-10 dataset.

In these models the CNLs were replaced by an IEA CNLs layers. The training was done using a non-augmented training samples. The training configuration is the same as the configurations mentioned in section 3.1.

Table 5 shows an overall improvement in classification mean error rates on CIFAR-10 dataset. VGG16 had a 0.64% mean error rate improvement by using IEA of CNLs. ResNet18 and ResNet101 were improved by 1.0%, 0.52% mean error rates respectively by using IEA compared to CNL. A significant improvement is seen in MobileNet by 3.09% error rate when using IEA, though IEA leads to an increase in total model run-time, which goes against the main purpose of MobileNet. DenseNet shows an improvement of 0.13% by using IEA of CNLs. To test the effect of models ensemble, table 6 shows an ensemble of previously mentioned models. We average an ensemble of three models using IEA of CNLs and another three models using CNLs only. The ensemble of IEA of CNLs models show a better performance than the CNL only models.

Table 5: Test set mean error rates in % with standard deviation on the CIFAR-10 dataset.

Deep model	CNL	IEA (ours)
VGG16	9.88± 0.16	**9.24± 0.29**
ResNet18	10.58± 0.14	**9.58± 0.02**
ResNet101	9.39± 0.20	**8.87± 0.34**
MobileNet	13.50± 0.05	**10.41± 0.23**
DenseNet	7.50± 0.07	**7.37± 0.10**

Table 6: Test set average ensemble error rates in % on the CIFAR-10 dataset.

Deep model	Ensemble of models using CNL	Ensemble of models using IEA (ours)
VGG16	8.36	**7.97**
ResNet18	9.26	**8.49**
ResNet101	7.69	**7.59**
MobileNet	10.95	**8.16**
DenseNet	6.31	**6.01**

4 VISUALIZATION AND ANALYSIS OF IEA FEATURES

To understand how IEA works and produces better results than ordinary CNL, we visualized the features generated by both. Figure 3 shows different features generated by IEA CNLs and CNL. It is noticeable that the features generated by IEA tend to be more unique and different from each other, unlike CNLs where some features appear to be identical. Also, in section (a) of figure 3 where the deep model is only 1 layer deep, there exist some IEA features that are associated with zero weight after training. This demonstrates that IEA removed some unnecessary features from the model, which helps improving robustness.

We use similarity scores λ to validate the visual interpretation of the IEA features. The similarity score measures how similar an image to another image. If both image of comparison are the same, the similarity score value will be zero. The more difference between the images the more the value of similarity score increases till it reaches the maximum value which is 1. Thus, the usage of similarity score is handful to measure how unique a feature compared to other features produced by the same layer. We introduce the mean sum of similarity score (mss-score) μ_λ as follows: for each n features f in a layer the mss-score is defined as:

$$\mu_\lambda = \frac{1}{n}(\sum_{i=1}^{n} \sum_{j=1, i \neq j}^{n} \lambda(f_i, f_j)) \tag{2}$$

The higher the mss-score is, the more unique the features produced by the model are. For the similarity score measurement, we used the similarity score model introduced by Zhang et al. (2018) which

is proven to outperform any previous similarity score measurements methods. In table 7, the IEA and CNL mss-scores were evaluated on a batch of 100 validation samples from the rotated-MNIST dataset. The IEAs mss-scores are always greater than the CNL mss-scores. This indicates that the usage of IEA produces more unique features compared to CNL features, and it confirms our visual analysis.

Table 7: The mss-scores of features generated by CNLs and IEAs. The score is measured for the first layer only in the models. The models were trained on rotated-MNIST dataset.

Case	CNL mss-score	IEA mss-score
1-layer model	0.034	**2.797**
2-layers model	0.020	**0.025**
3-layers model	0.020	**0.038**

5 Conclusion

IEA is a concept that is simple but powerful. It helps the CNN architecture to increase its prediction power by forcing the model to produce more unique features. The cost of using IEA in a CNN is on the model parameter size, yet it's not a significance cost to the improvement in the performance. We showed empirically, visually and by using a similarity scores that the usage of IEA improves the CNNs accuracy and produces unique features. Also, the ensemble of IEA of CNLs models outperforms the ensemble of CNL only model which is a method of inner and outer ensemble. We recommend the usage of IEA where it applies, and a further study of other methods of ensembles shall be conducted.

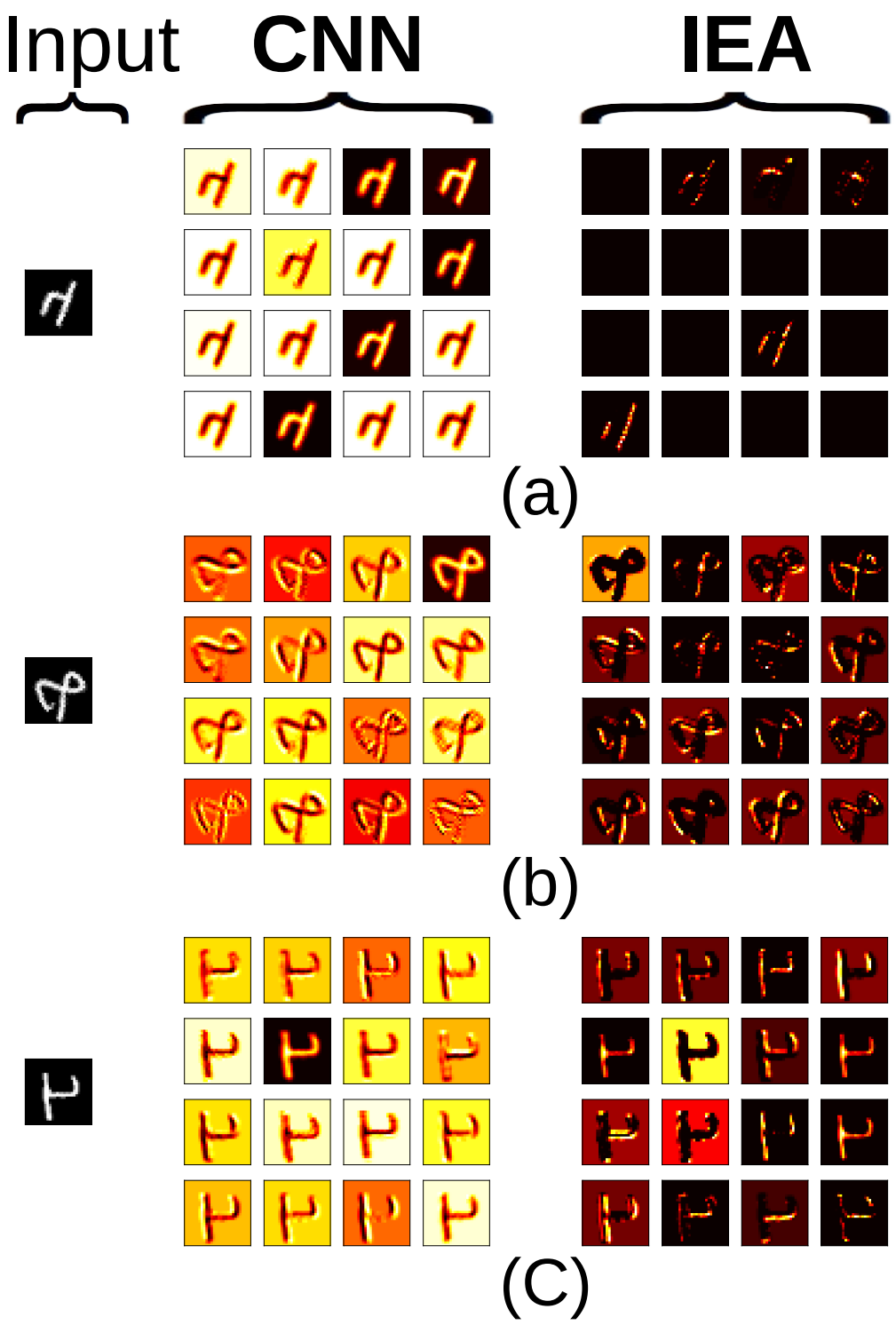

Figure 3: Features generated from both IEA and CNL. Section (a), (b) and (c) are the features of the first layer from one, two and three-layers deep models. The models were trained on the rotated-MNIST dataset. The input images are shown in a gray scale where the features are shown using a heat map color. The black color shade in the heat map indicates a minimum pixel value, where the white color indicates a maximum pixel value.

REFERENCES

Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *CoRR*, abs/1803.01271, 2018. URL http://arxiv.org/abs/1803.01271.

Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pp. 177–186. Springer, 2010.

Thomas G Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pp. 1–15. Springer, 2000.

Cicero dos Santos and Maira Gatti. Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pp. 69–78, 2014.

Harris Drucker, Corinna Cortes, Lawrence D Jackel, Yann LeCun, and Vladimir Vapnik. Boosting and other ensemble methods. *Neural Computation*, 6(6):1289–1301, 1994.

Stuart Geman, Elie Bienenstock, and René Doursat. Neural networks and the bias/variance dilemma. *Neural computation*, 4(1):1–58, 1992.

Richard HR Hahnloser, Rahul Sarpeshkar, Misha A Mahowald, Rodney J Douglas, and H Sebastian Seung. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature*, 405(6789):947–951, 2000.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL http://arxiv.org/abs/1512.03385.

Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017. URL http://arxiv.org/abs/1704.04861.

Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks. *CoRR*, abs/1608.06993, 2016. URL http://arxiv.org/abs/1608.06993.

David H Hubel and Torsten N Wiesel. Receptive fields and functional architecture of monkey striate cortex. *The Journal of physiology*, 195(1):215–243, 1968.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015. URL http://arxiv.org/abs/1502.03167.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

Anders Krogh and John A Hertz. A simple weight decay can improve generalization. In *Advances in neural information processing systems*, pp. 950–957, 1992.

Anders Krogh and Jesper Vedelsby. Neural network ensembles, cross validation, and active learning. In *Advances in neural information processing systems*, pp. 231–238, 1995.

Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. Recurrent convolutional neural networks for text classification. 2015.

Hugo Larochelle, Dumitru Erhan, Aaron Courville, James Bergstra, and Yoshua Bengio. An empirical evaluation of deep architectures on problems with many factors of variation. In *Proceedings of the 24th international conference on Machine learning*, pp. 473–480. ACM, 2007.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov 1998. ISSN 0018-9219. doi: 10.1109/5.726791.

Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Stefan Lee, Senthil Purushwalkam, Michael Cogswell, David J. Crandall, and Dhruv Batra. Why M heads are better than one: Training a diverse ensemble of deep networks. *CoRR*, abs/1511.06314, 2015. URL `http://arxiv.org/abs/1511.06314`.

David Opitz and Richard Maclin. Popular ensemble methods: An empirical study. *Journal of artificial intelligence research*, 11:169–198, 1999.

Boris T Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

Xueheng Qiu, Le Zhang, Ye Ren, Ponnuthurai N Suganthan, and Gehan Amaratunga. Ensemble deep learning for regression and time series forecasting. In *Computational Intelligence in Ensemble Learning (CIEL), 2014 IEEE Symposium on*, pp. 1–6. IEEE, 2014.

Lior Rokach. Ensemble-based classifiers. *Artificial Intelligence Review*, 33(1-2):1–39, 2010.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. URL `http://arxiv.org/abs/1409.1556`.

Felix Stahlberg and Bill Byrne. Unfolding and shrinking neural machine translation ensembles. *CoRR*, abs/1704.03279, 2017. URL `http://arxiv.org/abs/1704.03279`.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. *CoRR*, abs/1801.03924, 2018. URL `http://arxiv.org/abs/1801.03924`.

Zhi-Hua Zhou. *Ensemble methods: foundations and algorithms*. Chapman and Hall/CRC, 2012.

Zhi-Hua Zhou, Jianxin Wu, and Wei Tang. Ensembling neural networks: many could be better than all. *Artificial intelligence*, 137(1-2):239–263, 2002.