Research Seminar in Data Science
Presented Paper Review

Christoph Schmidl
s4226887
c.schmidl@student.ru.nl

June 17, 2018

- The paper **Dropout: A Simple Way to Prevent Neural Networks from Overfitting** by Srivastava et al. (2014) has been presented on the 17th of May 2018 by Christoph Schmidl during the Research Seminar in Data Science

1 Dropout: A Simple Way to Prevent Neural Networks from Overfitting

1.1 Objective

The main objective of this paper is to propose a method to prevent neural networks from overfitting. The so called "dropout" technique randomly drops units together with their connections in a neural network during training time and therefore prevents units from co-adapting. Co-adaptation seems to be one of the reasons that large networks tend to overfit on training data because these co-adaptations of neurons learn certain pathways in the network which are based on the training data but do not generalize well and therefore tend to have a higher error rate on the test set. The dropout technique samples so called "thinned networks" from the original network after dropping out or deactivating certain units randomly based on a certain probability value p. During test time these thinned networks get averaged to make predictions. Previous work to prevent overfitting incorporated techniques like early stopping critera during training, using regularization techniques like L1 and L2 regularization and soft weight sharing. Model combination also seemed to be a good technique but training different architectures and finding their optimal hyperparameters is a tedious task. Dropout should be a simpler regularization approach compared to the state-of-the-art techniques mentioned before. Therefore, the work is aimed towards machine learning practitioners who were seeking for an easy to use method to prevent overfitting in their neural networks.

1.2 Proposal made in the paper

The new idea presented in this paper is not only deactivating units during training in a neural network but that the random deactivation of units with a certain probability p leads to several thinned networks which later on get combined using an approximate averaging method. The idea of ensembling is not new. It has been done before by combining several weak predicitors of the same architecture which is called "bagging" Breiman (1996) with its most known implementation "Random Forest" Breiman (2001). Sampling subset networks of a bigger one and later on combining them again while they are still sharing the weights is a similar idea but another implementation on another domain, namely neural network structures. The paper claims that dropout is not solely applicable to feed forward networks but also to graphical models such as Boltzmann machines. Furthermore the authors show that dropout is applicable to different problem domains by using different kinds of datasets. It is shown that dropout successfully mitigates overfitting but nearly doubles the training time.

1.3 Evidence

The authors cite 36 different papers and support their assumption that dropout leads to more robust feature detectors by making a comparison to an evolutionary point of view Livnat et al. (2010) stating that mixability breaks up co-adapted genes and leads to more robust individuals. Furthermore, the authors state that dropout can also be seen as adding noise to the input layer because dropping hidden units randomly with a certain probability has a similar effect which makes it comparable to previous work based on autoencoders by Vincent et al. (2008) and Vincent et al. (2010). Previous work by the same authors also support the claim that dropout prevents neural networks from overfitting. The Master Thesis by Srivastava (2013) is used as the foundation of the work and the published paper which describes the usage of dropout to win the ImageNet Large-Scale Visual Recognition Challange (ILSVRC) of 2012 Krizhevsky et al. (2012) further hardens the claim. The following datasets have been used as benchmarks and show that dropout indeed improved the generalization error except for the Alternate Splicing data set where a bayesian network approach still outperforms a neural network with applied dropout.

- MNIST: A standard toy data set of handwritten digits
- TIMIT: A standard speech benchmark for clean speech recognition
- CIFAR-10 and CIFAR-100: Tuny natural images
- Street View House Numbers data set (SVHN): Images of house numbers collected by Google Street View
- ImageNet: A large collection of natural images
- Reuters-RCV1: A collection of Reuters newswire articles
- Alternate Splicing data set: RNA features for predicting alternative gene splicing

The authors do not only show that dropout performs well on most of the datasets, they also inspect the first layer of learned features and show the effect of dropout, namely that the learned features were more robust and sparse than the ones learned without dropout.

1.4 Shoulders of giants

One can argue that there are many ideas which influenced the main mechanics behind the ”dropout” technique but the following four papers represent the direct timeline in temporal order and foundation of the first occurrence of the word ”dropout” in the paper ”Improving neural networks by preventing co-adaptation of feature detectors” by Hinton et al. (2012) until the final paper ”Dropout: A Simple Way to Prevent Neural Networks from Overfitting” by Srivastava (2013). Each of the following subsections contains references to papers which are related to dropout or influenced its creation. Regardless of different ideas and inspirations to the dropout technique, the main authors of dropout seem to be:

- Geoffrey Hinton
- Nitish Srivastava
- Alex Krizhevsky
- Ilya Sutskever
- Ruslan Salakhutdinov

1.4.1 Improving neural networks by preventing co-adaptation of feature detectors

The paper ”Improving neural networks by preventing co-adaptation of feature detectors” by Hinton et al. (2012) has been published on the 3rd July 2012 and was cited 2746 times from publication until June 2018. This paper mentions ”dropout” for the first time and is written by the same authors as the ones in Srivastava et al. (2014) except that G.E.Hinton is mentioned as the first author instead of N. Srivastava. The dropout technique is presented as a way to prevent overfitting and complex co-adaptations. The paper describes the idea of dropping out ”feature detectors” or learning units during training with a certain probability incorporating L2 regularization as a constraint rather than a penalty term. During test time a ”mean network” is used which represents a kind of model averaging or ensemble method. The authors mention that the dropout technique is also applicable to Deep Boltzman machines and tested dropout on the following datasets:

- MNIST - Handwritten digits
- TIMIT - Recognition of speech with a small vocabulary
- CIFAR-10 - Benchmark for object recognition

- ImageNet - Challenging object recognition dataset with 1000 classes

After applying dropout to all networks which were trained on the before mentioned datasets, all of them had improved the generalization performance. According to the authors, dropout was inspired by several other ideas. Dropout can be seen as an extreme form of bagging Breiman (1996) where several weak predictors get combined to an ensembled model of predictors. Random forest is the most common implementation of this idea Breiman (2001). Another inspiration seems to come out of the domain of the theory of the role of sex in evolution Livnat et al. (2010) and that mixability breaks up co-adapted genes and leads to more robust individuals.
The authors showed for the first time that their proposed dropout idea is applicable to different problem domains and is increasing performance in all of them. They also incorporated an exhaustive appendix to the paper which contains more technical details, e.g. a visualization of features learned by first layer hidden units.

1.4.2 ImageNet Classification with Deep Convolutional Neural Networks

The paper "ImageNet Classification with Deep Convolutional Neural Networks" by Krizhevsky et al. (2012) has been published in 2012 and was cited 24405 times from publication until June 2018. It has probably been published after the paper Hinton et al. (2012) based on its citation. This paper has been cited nearly 9 times as much as the paper by Hinton et al. (2012). The popularity of this paper is probably based on the fact that the authors describe their approach of winning the ImageNet Large-Scale Visual Recognition Challange (ILSVRC) of 2012. Only a small part of the paper is about applying dropout but it is an essential part of their overall solution. Dropout is mentioned in subsection 4.2 of the section 4 "Reducing Overfitting" and is furthermore mentioned by the authors with "Without dropout, our network exhibits substantial overfitting. Dropout roughly doubles the number of iterations required to converge".
The proposed dropout technique probably got more attention after this paper because machine learning practitioners realized its benefits.

1.4.3 Improving neural networks with dropout

The master thesis "Improving neural networks with dropout" by Srivastava (2013) has been published in 213 and was cited 134 times from publication until June 2018. It cites the before mentioned papers by Hinton et al. (2012) and Krizhevsky et al. (2012) and adds or changes some minor details. Besides the MNIST and TIMIT dataset, this paper also describes the performance of dropout on the following datasets:

- SVHN - Images of house numbers collected by Google Street View
- Reuters-RCV1 - A collection of Reuters newswire articles
- Flickr-1M - Multimodal data (1 million images and tags)
- Alternate Splicing dataset - biochemistry data for genes

One interesting discovery is due to the experiments based on the alternate splicing dataset. The author compares his neural network approach which is using dropout with a Bayesian Neural Network. The Bayesian Neural Network outperforms the dropout approach which is not surprising because it was also known before that "Bayesian neural nets are extremely useful for solving problems in domains where data is scarce such as medial diagnosis, genetics, drug discovery and other bio-chemical applications". Furthermore, the author shows that the dropout idea works in the context of feed forwars neural networks and can be extended to Restricted Boltzmann Machines and other graphical models. Besides that the author also describes different effects of applying dropout. Learned features seem to become more sparse and robust because dropout can also be seen as some sort of enforced autoencoder similar to Vincent et al. (2008) and Vincent et al. (2010). Dropout only seems applicable to a sufficiently large dataset size because otherwise the network would suffer from underfitting. The same problem occurs when the hyperparamter p (dropout rate) is too low. P is the probability of keeping a feature detector or hidden unit and therefore when p is too low and the dataset is large, the amount of neurons is probably too low to learn any reasonable connection and therefore underfits.

1.4.4 Dropout: A Simple Way to Prevent Neural Networks from Overfitting

The paper "Dropout: A Simple Way to Prevent Neural Networks from Overfitting" by Srivastava et al. (2014) has been submitted in November 2013 and later on published in June 2014. It has been cited 6557 times from publication until June 2018. In comparison to the before mentioned papers and master thesis, this paper test dropout on the following datasets:

- MNIST: A standard toy data set of handwritten digits
- TIMIT: A standard speech benchmark for clean speech recognition
- CIFAR-10 and CIFAR-100: Tuny natural images
- Street View House Numbers data set (SVHN): Images of house numbers collected by Google Street View
- ImageNet: A large collection of natural images
- Reuters-RCV1: A collection of Reuters newswire articles
- Alternate Splicing data set: RNA features for predicting alternative gene splicing

Only one of the above datasets, namely CIFAR-100 has not been used before by the other papers. Therefore this paper does not add alot of knowledge in terms of used datasets compared to the other papers because it draws the exact same conclusions which have been made before. This paper seems to be a combination of the other papers and adds some minor details and restructures sections which have already been described earlier in the timeline. The only thing that seems to be new in this particular paper is the section about ”Multiplicative Gaussian Noise” and that the authors ”recently discovered that multiplying by a random variable drawn from $\mathcal{N}(1, 1)$ works just as well, or perhaps better than using Bernoulli noise.”
During the publication of the first paper about dropout Hinton et al. (2012) and this one, other authors proposed techniques for speeding up dropout by ”sampling from or integrating a Gaussian approximation, instead of doing Monte Carlo optimization of this objective. This approximation, justified by the central limit theorem and empirical evidence, gives an order of magnitude speedup and more stability” Wang and Manning (2013) and how dropout can be seen as an adaptive regularization technique by ”showig that the dropout regularizer is first-order equivalent to an L2 regularizer applied after scaling the features by an estimate of the inverse diagonal Fisher information matrix” Wager et al. (2013).

In my opinion, the paper ”Dropout: A Simple Way to Prevent Neural Networks from Overfitting” by Srivastava et al. (2014) is just a spinoff of papers by the same authors which have been published two years before. It does not add new knowledge but rather tries to describe the dropout technique with more detail and apply it to more datasets. Therefore this work could have been done way earlier and I do not see the need for publishing this paper because the Master Thesis by Srivastava already covered most the contents presented in this paper.

1.5 Impact

Based on the google scholar citations which show a total count of 6557 citations from June 2014 to now (June 2018), I would say that this paper had a major impact. If you want to restrict the time range and their corresponding citation counts to a more recent time, then google scholar gives the following information:

- Citations since 2017: 3960
- Citations since 2018: 1360

Based on these citations I would assume that dropout is still a relevant technique and that there have been work by other authors in order to improve dropout in terms of speed like Wang and Manning (2013). On the other hand there has also been work done which claims to make dropout obsolete or makes its usage less attractive. One of these techniques is batch normalization by Ioffe and Szegedy (2015) and the authors claim that ”dropout is typically used to reduce overfitting, in a batch-normalized network we found that it can be either removed or reduced in strength. By further increasing the learning rates, removing Dropout, and applying other modifications afforded by Batch Normalization, we reach the previous state of the art with only a small fraction of training steps and then beat the state of the art in single-network image classification.”

1.6 Discussion

The following questions have been send to me via Email and have been discussed during the presentation:

- How do you think Dropout compares to other techniques to prevent overfitting, like L2 regularization and Batch Normalization?

  - Dropout is using L2 regularization as a constraint rather than a penalty term, therefore it is querionable if additional L2 regularization is needed when dropout is already applied. Batch Normalization seems to be a more recent alternative to dropout and is also faster during training. One has to decide for himself if it makes more sense to apply dropout or batch normalization.

- When should Dropout be used over other techniques?
  - Dropout can only be applied to neural network like structures. Therefore the application domain is already restricted. When the dataset is small then dropout does not seem to make much sense because it leads to an underfitting network. Other regularization techniques which are less extreme seem to make more sense in this case.

- Should we always use Dropout?
  - Like stated in the previous discussion point we should not use dropout all the time. It depends on the network structure and the size of the dataset. The paper nicely describes the performance based on data set size and the number of hyperparameters.

- Do you think the length of the paper was necessary to make a compelling case for the intrusive regularization method?
  - I have to admit that the paper was unnecessary long. Most of the work has already been presented in other papers or master thesis. This paper seems to be a bigger summary of the work that has been done to this point. So "no", it was not necessary.

- It appears that Dropout is not used anymore in a number of large available networks such as Resnet, InceptionV3. Why do you think they do not use Dropout?
  - Newer architectures rather seem to use batch normalization instead of dropout because it also mitigates co-variate shift.

- Given the paper Understanding Deep Learning Requires Rethinking Generalization from the last session, do you think that dropout actually prevents overfitting or might there be another reason why the generalization error is decreased?
  - That is hard to tell. Based on the benchmarks presented in this paper there seems to be a direct connection between the application of droput and the prevention of overfitting.

- Newer articles hardly ever mention dropout anymore. Many deep supervised models are nowadays done with residual convolutional architectures that make use of batch normalization. Has dropout become redundant?
  - Dropout is still used in some cases nowaday if you look up the citation count on google scholar. But it is true that most people switch over to batch normalization because it also holds some properties which dropout does not have like mitigating co-variate shift. Another reason that newer articles and architectures are not using dropout is based on the architecture of the network itself. Newer architectures probably tackle the problem of overfitting by their structure itself so that dropout does not provide any benefits.

- Since the authors note that removing any convolutional layer resulted in inferior performance, can we expect better performance if we add even more convolutional layers?
  - That probably depends on the type of dataset and its size. If the network is already able to learn the underlying distribution of the dataset with its hyperparameter count then adding additional layers only increases overfitting.

- A hyper parameter p with a high value can be interpreted as resulting in too complex co-adaptations between neurons. What could be an interpretation of a higher error associated with a low p value?
  - Underfitting because a low p value indicates that only a small fraction of the network units are retained. If the amount of neurons is so small that it cannot learn the underlying distribution of the dataset in a sufficient way, then this represents the idea behing underfitting.

- The authors mention a speedup of Dropout is an interesting direction for future work. Have people tried improving Dropout and how much did it improve?
  - Wang and Manning (2013) were able to speed up dropout by an "order of magnitude" by "sampling from or integrating a Gaussian approximation, instead of doing Monte Carlo optimization".

- The authors mention that the gradients that are being computed are not gradients of the final architecture. This increases training time as the network is effectively training a different network every time. Do you think there's a solution for this limitation? Perhaps there's some sort of technique for optimizing these gradient updates such that the training time decreases.
  - This discussion point was not answered. I looked this point up but could not find any answers. One student mentioned that batch normalization is targeting this exact problem.
- The authors mention that multiplicative Gaussian noise led to equal or better performance when the mean and variance matched a Bernoulli distribution. However, they do not provide any explanation for this finding. Do you have an idea of why this might be better?
  - We did not reach this discussion point during the presentation and therefore had to omit it.

References

Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2):123–140.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc.

Livnat, A., Papadimitriou, C., Pippenger, N., and Feldman, M. W. (2010). Sex, mixability, and modularity. *Proceedings of the National Academy of Sciences*, 107(4):1452–1457.

Srivastava, N. (2013). Improving neural networks with dropout. Master's thesis, University of Toronto, Canada.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958.

Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A. (2008). Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., and Manzagol, P.-A. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(Dec):3371–3408.

Wager, S., Wang, S., and Liang, P. S. (2013). Dropout training as adaptive regularization. In *Advances in neural information processing systems*, pages 351–359.

Wang, S. and Manning, C. (2013). Fast dropout training. In *international conference on machine learning*, pages 118–126.