Exploring the Variations of Dropout

Yunfei Luo
University of Massachusetts Amherst
yunfeiluo@umass.edu

Meredith Pan
Smith College
mpan@umass.edu

Abstract

Deep Neural Networks had shown to be powerful on the task of image classification. However, it faces a very serious problem – overfitting, when dealing with a large amount of parameters. To avoid overfitting, many past researches implement dropout for regularization. In this project, we proposed several variations of Dropout. We provided performance comparison of different methods of Dropout on three benchmark data sets: MNIST, SVHN, and CIFAR-10/100. The experimental results reveal that our methods could improve the performance beyond the baseline.

1. Introduction

Our research follows the idea of Dropout [6] and investigates its variations.

Overfitting is a commonplace problem in training neural networks. Various techniques were introduced to address this problem and dropout [6] have been among the popular ones. It is demonstrated to be useful in terms of both regularization and feature selection. The Dropout process can be seen as training different models during training, and greedily averaged the output during the testing. However, because dropout is a fully stochastic process, where the neurons are clipped to zero with a fixed probability that is determined as a hyper-parameter, the process can be uncontrollable and can drop some useful features during training. In order to address this problem, we propose three variations of dropout.

The first variation is Leaky Dropout inspired by Leaky ReLU [2] where the dropped neurons will be clipped with a slope factor instead of zero.

The second variation called Softmax Weighted Dropout is based on the idea of Weighted Channel [4] where the dropped probabilities are determined by the significance of the neurons across the entire features map. The fact that it drops more significant data points with higher probability has its rationale in stochastic max-drop technique [3], which selectively drops the maximum activation within each feature map or within the same spatial position of feature maps.

Finally, the third variation is called Gumbel-Trick Dropout that is inspired by [1], where the neuron will either be or not be dropped according to a binary selection drawn with Gumbel-Trick process. These variations aim to provide regularization and achieve better predictions on classification tasks.

2. Related work

In recent years, Convolutional Neural Networks(CNN) have gained great popularity in the field of computer vision, including image classification tasks. However, it has a risk of overfitting, especially in the case where there is a small training data set. So the Dropout [6] method was proposed and is one of the most commonly used techniques to prevent overfitting. It requires a hyper-parameter p that represents the probability of dropping each neuron. This random dropping process enables the network to train only parts of itself during each training iteration and to greedily take their average during testing. This makes sure that the model does not overfit by preventing simultaneous optimization of all neurons. It has shown to be effective in various implementations in deep neural networks. However, the fact that it assigns the same drop-out probability p to every neuron may ignore the difference in neurons' influence on the prediction. And thus the network may train without essential features or with many unhelpful ones. Such problem is the essential motivation of our proposed variations.

Leaky ReLU [2], as an inspiration of the Leaky Dropout we proposed, aims to solve the problem of killing the gradients, also known as "dead ReLU" in traditional ReLU, where the incoming values are 0 and so always obtain 0 gradients and not learn. The leaky version prevents the gradients being killed by adding a small positive slope/weight on the negative values, which is pre-determined as a hyper-parameter. Empirically, this method loosens the non-negativity constraints on weight initialization.

Another variation that we purpose is a Softmax Weighted Dropout(SWD). It is inspired by Weighted Channel Dropout(WCD) [4], which drops the channels with a probability according to their incoming values. And different

from Dropout that acts on individual neurons, WCD aims to regularize convolutional layers in CNN. Specifically, WCD rates the channels and assigns each of them a score according to their activation status, and then use this score as the channel' dropping probability. SWD that we propose, on the other hand, follows a similar idea of assigning scores to data in correspondence to their incoming values. But its main differences from WCD are that it is not only applied on the convolutional layers but also fully connected ones and that the scores are calculated by softmax function with an additional temperature parameter that controls its slope. When applied to fully connected layers, SWD will calcualte the relative significance of each individual neurons among features, and when applied to convolutional layers, SWD will calculate that among channels. This extends the implementation of weighted dropout further to fully connected layers. The rationale of using softmax function in calculating the scores has its root in Max Drop [3], which drops the most significant feature. It is shown to work well with problems such as occlussion and distortion in image classification tasks, where the most significant features are not always necessary. In SWD, the softmax function is an increasing function and will maintain the relative difference between neurons. The larger the features are, the higher their softmax score will be. And as Max Drop drops exactly one most significant feature, the expected number of dropped feature after softmax function is also exactly one.

Finally, Gumbel-Trick Softmax [1] is the core of our third proposed variation. The issue in our proposed Gumbel-Trick Dropout is that the dropping criterion is a maximum operation on the scores assigned to the binary choice of either drop or not drop. The process is not differentiable. And it is also not rigorous to make the scores as hyper-parameters. Gumbel-Trick Softmax was initially used for smoothing the operation of making choice from a discrete space. It makes the maximum operation differentiable. As a result, when the Gumbel-Trick Softmax is adopted on making choice from the binary scores, those scores becomes trainable parameters. Under such setting, the model will learn whether to drop or keep each neuron during training. During test time, the original dropout greedily takes average from different sub-parts of the model, while Gumbel-Trick Dropout aims to take the best part from the model.

3. Methodology

3.1. Variations of Dropout

3.1.1 Leaky Dropout

Instead of clipping the dropped neuron straightly to zero, the leaky dropout technique adds a slope factor instead. It requires two hyper-parameters: p, the probability of dropping the neuron, and k, the slope on dropped neurons. At training time, we scale each neuron by k with probability p and otherwise directly pass it to the next layer. Since at test time the neuron will be passed along unchanged, we keep the consistency of expected value at test and train time by dividing each neuron by $(pk+1-p)$ during training. Thus,

$$\text{LeakyDropout}(x) = \begin{cases} kx/(pk+1-p) & \text{, with probability } p \\ x/(pk+1-p) & \text{, otherwise} \end{cases}$$

3.1.2 Softmax Weighted Dropout

In softmax weighted dropout, the probability of dropping each neuron is determined by its importance over entire feature maps. Denote X as a $N \times D$ input matrix, where N is the batch size, and D is the feature dimension. Then the dropout probability $p_{i,j}$, where $i \in \{0, 1, ..., N\}, j \in \{1, 2, ..., D\}$, is determined by

$$p_{i,j} = \frac{\exp(X_{i,j})/\tau}{\sum_{k=0}^{D} \exp(X_{i,k})/\tau}$$

where τ is a temperature hyper-parameter that controls the slope of the softmax function.

In this layer, the more significant features are more likely to be dropped. This is closely related to the idea of stochastic max-drop, which shows that dropping the most influential feature can avoid problems such as oscillation and distortion, and since the expected number of dropped neuron is 1 among X_i in softmax weigted dropout layer, this method can be regarded as a stochastic variation of max-drop.

3.1.3 Gumbel-Trick Dropout

The Gumbel-Trick dropout introduces two additional parameters, θ_0 and θ_1, for each neuron. The output through Gumbel-Trick dropout on an incoming neuron x is expected to be

$$x' = x \times [1, 0] \cdot one\_hot\{\arg\max_i(\log(\theta_i))\}$$

where $i \in \{0, 1\}$.

If for a neuron x, $\theta_0 < \theta_1$, then $[1, 0] \cdot one\_hot\{\arg\max_i(\log(\theta_i))\}$ will result in 0,k and thus x will be "dropped out". Otherwise, x will be passed to the next layer.

In order to ensure that the network does not always train the same larger of the two parameters θ_0, θ_1 for each neuron and that $one\_hot\{\arg\max_i(\log(\theta_i))\}$ is not always the same, we add a noise term ϵ during training time. Thus, the expected output through Gumbel-Trick dropout on an incoming neuron x during training is:

$$x' = x \times [1, 0] \cdot one\_hot\{\arg\max_i(\log(\theta_i) + \epsilon_i)\},$$

where ϵ is a noise term drawn from a normal distribution with zero mean and unit standard deviation.

Lastly, we want to make this operation differentiable in training time, we follow the instruction in [1], and we apply a softmax operation in replace of the one-hot vector, i.e,. the one-hot vector $one\_hot\{\arg\max_i(\log(\theta_i))\}$ is approximated by the vector

$$\frac{\exp((\log(\theta_i)+\epsilon_i)/\tau)}{\sum_{j\in[0,1]}\exp((\log(\theta_j)+\epsilon_j)/\tau)}$$

where τ is a temperature that controls the slope of approximation. And formally, the output through Gumbel-Trick dropout on a neuron x is:

$$x' = x \times [1,0]\cdot\frac{\exp((\log(\theta_i)+\epsilon_i)/\tau)}{\sum_{j\in[0,1]}\exp((\log(\theta_j)+\epsilon_j)/\tau)}$$

3.2. Model structure

3.2.1 Fully Connected Neural Network

For the fully connected neural network, we follow the Standard Neural Network [5] as used in first paper on Dropout [6]. Standard Neural Network is composed of 3 fully connected layer with ReLU as activation function.

3.2.2 Convolutional Neural Network

For the convolutional Neural Network, we follow the LeNet-5 setting as used in [6]. LeNet-5 consists of three 5×5 convolutional layers with stride 1, followed by two fully connected layers with 2048 neurons for each. There is a max-pooling layer after each convolutional layer, and ReLU is used for all activation.

3.3. Evaluation

We evaluate the models on its performance on image classification tasks on test set. The evaluation incorporates 3 folds: accuracy, F1 score and AUC score.

3.3.1 Accuracy

Accuracy of the result is calculated as the percentage of correct classification results among all results, i.e, the number of correct results divided by the total number of results.

3.3.2 F1 score

We use macro F1 score for evaluation. And it is calculated as the average of F1 scores over all labels.

3.3.3 AUC score

We use weighted AUC score. It is the weighted average of AUC scores over all labels, where AUC score for an individual label represents the probability that a random example with the label has a higher score than a random example with a different label.

4. Results

We trained neural networks with dropout variations for classification problems on image data sets. The data sets are:

- MNIST: a dataset of handwritten digits
- Street View House Numbers (SVHN): a real-world image dataset of house numbers in Google Street View images
- CIFAR-10/100: tiny natural labelled images

Table 1 gives an overview of the data sets.

Data Set	Dimensionality	Train Set	Test Set
MNIST	28 ×28 grayscale	60K	10K
SVHN	32 ×32 color	600K	26K
CIFAR-10/100	32 ×32 color	60K	10K

Table 1. Overview of data sets.

We use mini-batch Nesterov SGD with 0.9 momentum and L2 regularization with coefficient of 1e-4 for training all the models. We fixed the batch size to 128, and trained each model with 20 epochs. The initial learning rate for most of the models is 1e-3, with a decay factor of 0.1 at 50% and 75% of total epochs.

4.1. MNIST

The MNIST data set is composed of 28×28 grayscale images of handwritten digits. The task is to classify the images according to digits 0-9.

Table 4 presents a comparison of the performance of the three dropout variants with other techniques. Since image classification task on MNIST data is relatively simple given that it contains only grayscale images that can be represented with a single channel, we use fully connected neural network for the task. We can see that networks implementing Leaky Dropout, Softmax Weighted Dropout and Gumbel-Trick Dropout outperform the ones with tranditional dropout technique. And Gumbel-Trick Dropout achieves the lowest error rate among the three.

The baseline neural network is composed of 3 fully connected layers with 1024 units and uses ReLU for all activations. It achieves an accuracy of 96.7%. Adding dropout improves the accuracy to 97.4%. And the three varaints of dropout that we propose further inhanced the result. Adding

Method	Accuracy	F1	AUC
Standard Neural Net	0.967	0.966	0.982
NN + dropout	0.974	0.974	0.986
NN + leaky	0.976	0.976	0.987
NN + softmax	0.978	0.978	0.988
NN + gumbel	**0.980**	**0.980**	**0.989**

Table 2. Performance on MNIST data set.

leaky dropout in the baseline model achieves an accuracy of 97.6%; adding weighted softmax dropout achieves an accuracy of 97.8%; and finally, adding gumbel dropout achieves the highest accuracy of 98.0%, which obtains a 1.3% increase in accuracy compared to the baseline model.

4.2. Street View House Numbers (SVHN)

The Street View House Numbers (SVHN) Dataset is a dataset with real-world colored images of house numbers collected by Google Street View. The image classification task is to identify and classify images according to the digit that is roughly in the center of the house number image.

Table 4.1 compares the performance of our dropout variants with other methods on SVHN. Since images in SVHN have rgb colors, which requires 3 channels to represent, the image classification tasks on it is usually done with CNN as it can exact features from images efficiently. In [6], the authors shown that adding dropout on all layers can improve the performance, but it is not the case for some experiments we had conducted. As a result, we experimented with adding dropout and its variants layers into all layers as well as only fully connected layers. The results shows that the three dropout variations achieves similar error rates with tradition dropout, and that dropout and the variants seems to work better on fully connected layer only.

The baseline model follows the LeNet-5 setting and provides an accuracy of 91.3%. And adding dropout to fully connected layers boosts the accuracy to 91.5%. We experimented with the three variants by adding them in all or only fully connected layers. The best results are given by the baseline model with leaky dropout, weighted softmax dropout or gumbel dropout on fully connected layer, which achieves an accuracy of 91.5%, 91.4% and 91.4% respectively.

4.3. CIFAR-10

The CIFAR-10 data set is one of 32×32 colored images that are labelled by 10 classes: airplane, automobile, bird, cat, deer, dog, frog, horse, ship and truck.

Table 4.1 presents the evaluation of dropout variants' performance on CIFAR-10. The baseline model that follows the structure of the LeNet-5 network [6] provides an accuracy of 74.8% without dropout, and 74.6% with traditional dropout on all fully connected layers. Adding weighted softmax dropout to all layers and adding gumbel dropout to fully connected layer both improves model's performance. The former achieves an accuracy of 74.9%, boosting the accuracy by 0.1% from baseline model and by 0.3% from baseline model with traditional dropout. The later provides the best performance of all models we experimented with. It achieves an accuracy of 75.3%, which shows a 0.5% improvement from the baseline model.

4.4. CIFAR-100

The CIFAR-100 data set contains the same images as in CIFAR-10. Instead of labeling the natural images into 10 classes, CIFAR-100 labels them into 100 classes.

As presented in Table 4.1, the baseline model with LeNet-5 structure [6] achieves an accuracy of 43.8%. An additional weighted softmax dropout in all layers as well as fully connected layers provide better results, achieving 44.2% and 44.3% accuracy respectively. An additional gumbel dropout in fully connected layer also increases the accuracy and improves it by 1.4%, giving a 45.2% accuracy.

5. Analysis and Discussion

The experiment leads to two interesting results.

First, gumbel dropout in fully connected layer behaves well when added to both standard neural network [5] and LeNet-5 network [6]. It achieved at best an improvement of 0.6% on standard neural network, and an improvement of 6.9% on LeNet-5 network compared to them implementing traditional dropout method.

Second, although dropout and its variants seem to perform worse when added to all layers than when added to only fully connected layers, softmax weighted dropout performs well on both. When the dropout and its variations are added to the fully connected layer only, the convolutional layers are still being regularized as well because of the zero output neurons at the final layer before computing the classes scores. When the dropout and its variations are added to all the layers, the strength of regularization is raised directly, since more neurons could be dropped. The reason that the performance is worse when adding dropout and the leaky version to all layers could be the problem of under-fitting because of the strong regularization. With more neurons being dropped during each iterations, there are more sub-part models being trained, which directly increase the variance of the output. The large variance of the output during training could directly lead to difficulty in convergence. As we know that dropout can be seen as greedily taking average across all the sub-part being trained, the difficulty in convergence make it hard to improve the performance on the test set. With Gumbel-Trick Dropout, when adding to all layers, the number of parameters that will be used for prediction is reduced. As Gumbel-Trick Dropout will use the best sub-part of the model in testing

	SVHN			CIFAR-10			CIFAR-100		
Models	Acc.	F1	AUC	Acc.	F1	AUC	Acc.	F1	AUC
ConvNet	0.913	0.913	0.951	0.748	0.748	0.860	0.438	0.438	0.716
ConvNet + dropout all layers	0.881	0.880	0.933	0.588	0.593	0.777	0.082	0.049	0.536
ConvNet + dropout f.c. layers	**0.915**	**0.915**	**0.952**	0.746	0.744	0.857	0.383	0.368	0.688
ConvNet + leaky all layers	0.885	0.885	0.935	0.592	0.588	0.773	0.094	0.060	0.542
ConvNet + leaky f.c. layers	**0.915**	**0.915**	**0.952**	0.743	0.742	0.857	0.401	0.387	0.697
ConvNet + softmax all layers	0.914	0.914	**0.952**	0.749	0.749	0.860	0.442	0.442	0.718
ConvNet + softmax f.c. layers	0.913	0.913	0.951	0.745	0.746	0.858	0.443	**0.443**	0.719
ConvNet + gumbel all layers	0.894	0.894	0.940	0.648	0.634	0.804	0.262	0.233	0.627
ConvNet + gumbel f.c. layers	0.914	0.914	**0.952**	**0.753**	**0.751**	**0.863**	**0.452**	0.441	**0.723**

Table 3. Results on the Amazon product review dataset. "FC-Net" denotes the fully connected neural network; "ConvNet" denotes the convolutional neural network; "s" and "d" stand for "shallow" and "deep," respectively; and "w2v" and "char" stand for Word2Vec and character level embeddings, repectively.

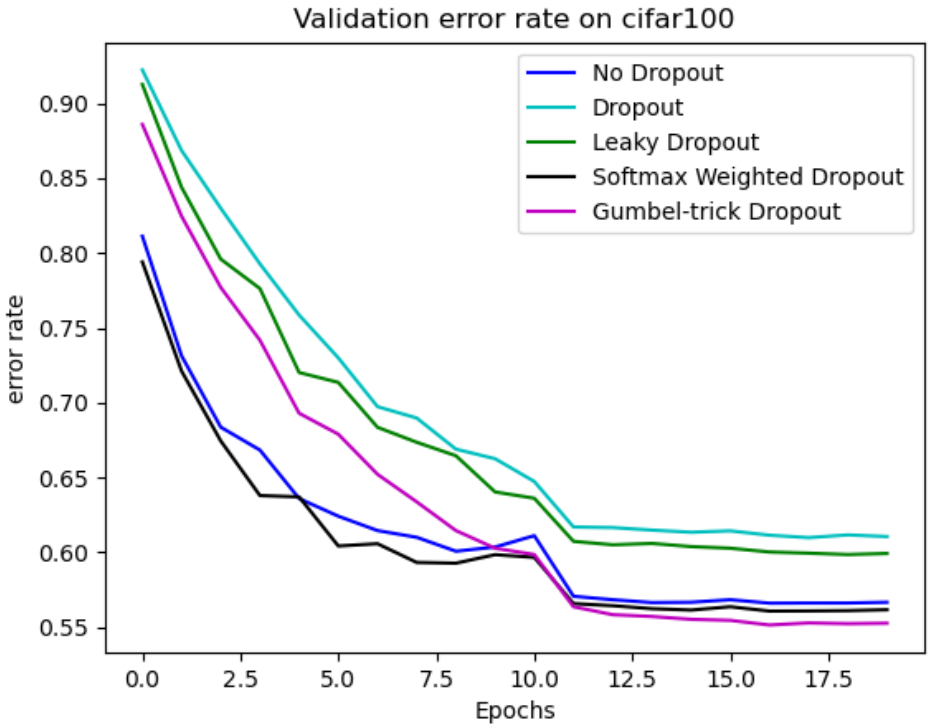

Figure 1. Validation error rate on CIFAR-100

Method	#Params
NN + dropout	∼1.864M
NN + gumbel	∼1.868M
ConvNet + dropout f.c. layers	∼5.877M
ConvNet + gumbel f.c. layers	∼5.885M

Table 4. Number of Parameters.

time, less parameters could lead to insufficiency during feature extraction.

For the aspect of efficiency, the variations of dropout we had proposed didn't add extra parameters except the one with Gumbel-Trick. As shown in Table 4, there are about 10k parameters added when the Gumbel-Trick layers are introduced after all the fully connected layers. The parameters added by adopting Gumbel-Trick Dropout is linearly related to the number of features in the feature maps of each

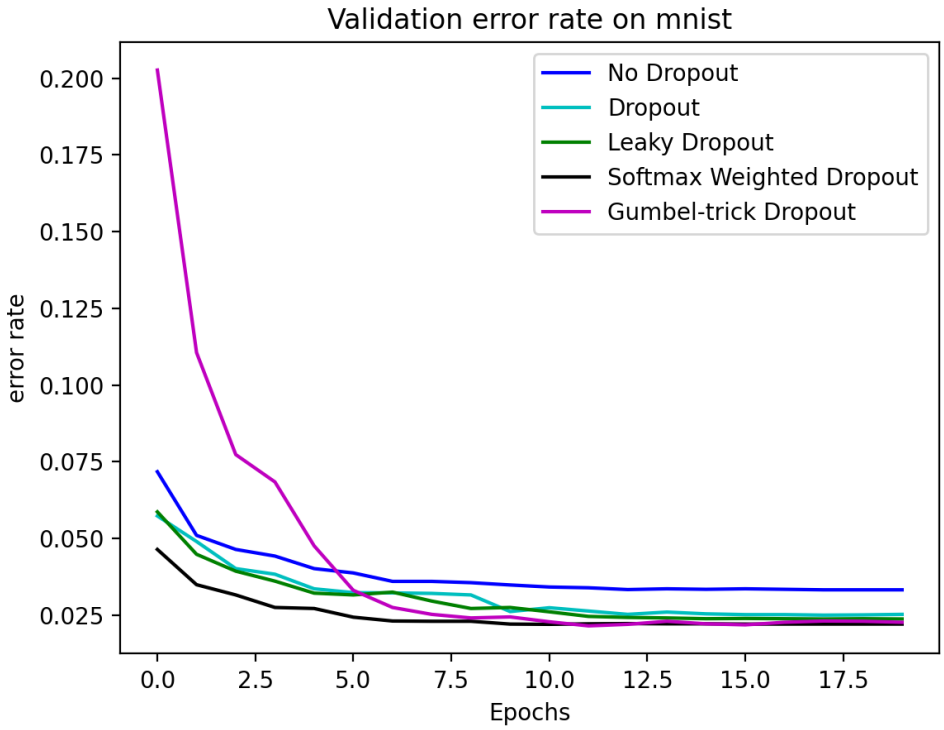

Figure 2. Validation error rate on CIFAR-100

layer. Comparing to the number of parameters in the filters' weights, there is only a tiny offset being added.

In addition, we inspect the influence of the dropout variations on the reduction of validation error rates. Figure 1 shows the validation error curve on CIFAR-100. We can see that after all 20 epochs, Gumbel-trick dropout and Softmax-Weighted dropout perform better than without any dropout. The original dropout and leaky dropout seems to be losing the battle. Adding dropout could directly mitigate overfitting, but dropout and Leaky dropout are sensitive to the initialization of dropping probabilities. It is highly likely that the regularization might not be efficient. While Softmax-Weighted dropout and Gumbel-trick dropout will adjust the dropping probabilities to adapt the refreshing situations of the model output during the training. They are more reasonable methods. Figure 2 tells a similar story, where Gumbel-trick dropout and Softmax-Weighted dropout achieves the lowest error rate. Notice that although the initial error rate in the model implementing Gumbel-trick dropout is relatively large in this case, it adapts and reduces error rate fast.

When we inspect the strength of regularization, we found that Softmax-Weighted Dropout will drop less number of parameters than other variations, since the algorithm expect to drop only one feature at each layer. This could explain why the performance of Softmax-Weighted Dropout is close to the model without any dropout. For Gumbel-Trick Dropout, the best sub-part of the model is greedily selected during test time. While the original dropout and leaky dropout will use all the sub-parts for averaging, where the relatively large variance could negatively influence the performance.

6. Conclusion and Future work

In this project we proposed three variations of dropout: Leaky Dropout, Softmax-Weighted Dropout, and Gumbel-Trick Dropout. They are aiming to address the problems of having uncontrollable stochastic process and useful features being dropped with the original dropout operation. We test our methods on the image classification task. The empirical results indicate that the variations we proposed could not only provide more reasonable regularization, but also improve the performance beyond the baseline.

For the future work, we hope to see that the variations of dropout we proposed could also provide efficient regularization on other tasks such as localization and NLP tasks, while still bring improvements to the performance.

References

[1] E. Jang, S. Gu, and B. Poole. Categorical reparameterization with gumbel-softmax, 2017. 1, 2, 3

[2] A. L. Maas. Rectifier nonlinearities improve neural network acoustic models. 2013. 1

[3] S. Park and N. Kwak. Analysis on the dropout effect in convolutional neural networks. In S.-H. Lai, V. Lepetit, K. Nishino, and Y. Sato, editors, *Computer Vision – ACCV 2016*, pages 189–204, Cham, 2017. Springer International Publishing. 1, 2

[4] H. Saihui and W. Zilei. Weighted channel dropout for regularization of deep convolutional neural network. In *AAAI Technical Track: Vision*, pages 1–30. AAAI Conference on Artificial Intelligence (AAAI)., 2019. 1

[5] P. Simard, D. Steinkraus, and J. C. Platt. Best practices for convolutional neural networks applied to visual document analysis. *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.*, pages 958–963, 2003. 3, 4

[6] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. 1, 3, 4