ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

Constraints as Alternative Learning Objective in Deep Learning

Quinten Van Baelen[1,2] and Peter Karsmakers[1,2] *

1 - KU Leuven, Geel Campus, Dept. of Computer Science;
Leuven.AI, B-2440 Geel, Belgium

2- Flanders Make@KU Leuven, Belgium

{quinten.vanbaelen,peter.karsmakers}@kuleuven.be

Abstract. The success of deep learning has been based on smooth loss functions that can easily be optimized using gradient descent and an off-the-shelf optimizer. However, training a neural network for a new application is not trivial as it requires many hyperparameters to be tuned. Several issues exist such as overfitting and underfitting. Many applications allow for some errors to be made, although, traditional learning objectives will influence the training in all cases except the one perfect prediction is made. In this work, constraints are proposed to replace the cross-entropy or the mean squared error to allow the neural network to make some errors. These errors can be set in advance to reflect how accurate the predictions of the neural network need to be. For each loss function, it is shown on two different data sets that the proposed constraint based learning performs similarly or even outperforms the standard loss functions. Moreover, in the case of classification problems, the constraints can result in predictions with significantly higher probability on a test set.

1 Introduction

Deep learning has been successfully applied in a wide range of applications in recent years. These successes occur mostly in applications where plenty of data is available. However, for several applications only a limited amount of data is available. When applying deep learning to cases where data is scarce, suboptimal generalisation abilities are expected. For example, a large neural network can easily overfit on a small training set. Moreover, neural networks are typically trained using gradient descent or a variant which require learning objectives to be smooth. However, smooth objectives might only approximate the exact target as is defined by end-users. For example, in the context of semantic segmentation, there might be several segment boundaries that slightly differ from each other which are all equal from a user point of view. However, a loss function does not, typically, assign the same loss value to all these predictions. Another example, in the context of multi-class classification problem, a neural network is expected to provide a single label for the input. It is not crucial in the standard setting with which probability this label is predicted. These examples show that the neural network does not need to predict the exact groundtruth label but rather

*This research received funding from the Flemish Government (AI ResearchProgram). This research has received support of Flanders Make.

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

a prediction which is sufficiently close. This is closely related to improving the probability of a neural network to predict a plausible prediction [1].

We propose to train a neural network with non-smooth objectives that are more closely related to the user requirements for a given application. These non-smooth objectives are formulated as constraints, leading to a constraint satisfaction problem (CSP) for the weights of the neural network. Previous work [2] exists that aim at satisfying constraints, in the context of CSPs, in combination with a traditional loss function, but no method has been applied to train a network completely by solving a CSP. Moreover, DeepSADE [2] uses only the CSP to train the final layer of a neural network and not the full network. In this text, a direct comparison is made between the usage of a loss function in an optimization problem and a CSP for training a neural network.

Our main contributions are: (i) the reformulation of a loss based neural network learning approach to a CSP which is solved by Constraint Guided Gradient Descent (CGGD) [3] such that the learning objective is defined more closely to that defined by the user, (ii) an extensive experimental evaluation which indicates that the proposed CSPs obtain similar or better results compared to traditional loss functions.

The remainder of this text is structured as follows. The set of constraints that can be used as alternative to some traditional loss functions are described in Section 2. In Section 3, an experimental evaluation of the proposed objectives compared to the traditional loss functions are described. Afterwards, the results are shown and discussed in Section 4. Finally, the text is concluded with possible directions for future work in Section 5.

2 Constraint based learning

Within this section, a supervised multi-class classification problem and a supervised regression problem are considered for a neural network. For the supervised multi-class classification problem, it is common to use an output layer with softmax activation function and use Cross-Entropy (CE) as a loss function. The softmax activation function results in the output of the neural network defining a probability distribution. Therefore, if a single output neuron has a value strictly larger than 0.5, then the corresponding class would be the networks prediction for a Top-1 accuracy. In other words, if there are N possible classes and x is an example of class C_i, then it is sufficient to find weights $\boldsymbol{W}$ of a neural network $\boldsymbol{\Phi}$ such that $\boldsymbol{\Phi}(x)_i > 0.5$, where $\boldsymbol{\Phi}(x)_i$ is the value of the i-th output neuron corresponding to class C_i. Moreover, the negative classes can also be required to have a relatively small probability. For example, the predicted probabilities for a sample of class C_i should be below 0.1 for all classes C_j with $j \neq i$. This leads to the following set of constraints for the CE with variable bounds T and F

$$\forall x \in \boldsymbol{X} : \boldsymbol{\Phi}(x)_{C_i} > T \text{ and } \boldsymbol{\Phi}(x)_{C_j} < F, \quad \text{for } j = 1, \ldots, N, \text{ and } j \neq i. \tag{1}$$

A similar reasoning can be performed for a supervised regression problem,

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

which is often trained using the Mean Squared Error (MSE). Indeed, a relatively small deviation of the ground truth is allowed. Therefore, the weights should be changed only in the case where the difference between the prediction and the groundtruth is larger than some threshold. For example, an alternative set of constraints for the MSE is

$$\forall x \in \boldsymbol{X} : \boldsymbol{\Phi}(x) \in (y - \delta, y + \delta), \tag{2}$$

where y is the groundtruth of x, and $\delta > 0$ is the number defining the allowed deviation of $\boldsymbol{\Phi}(x)$ from y.

The problems should be solved by finding a suitable set of weights $\boldsymbol{W}$ such that all constraints are satisfied on the training set. This requires an algorithm that can check these constraints and adjust the weights such that eventually the predictions should satisfy the constraints. The performance metric used to check how many constraints are satisfied is the Satisfaction Ratio (SR) [3], which is the ratio of the number of satisfied constraints by the total number of constraints. The CGGD [3] framework supports this setting as it solves constrained optimization problems, even in the case where no objective function is present but some constraints are. In other words, CGGD can also be applied to CSPs. In order to apply CGGD, it is required to define the direction of the constraints $dir_{\boldsymbol{C}}$ for each constraint. However, all constraints above are bound constraints as they define a lower or upper bound on some prediction of the network. When a lower bound is not satisfied, then $dir_{\boldsymbol{C}}$ can be set to -1 as this will increase the prediction after a gradient descent update. When a upper bound is not satisfied, then $dir_{\boldsymbol{C}}$ can be set to 1 as this will decrease the prediction after a gradient descent update.

3 Experiments

The supervised multi-class classification task is evaluated on MNIST and CIFAR10. The publicly available validation sets are used as test sets and the best network during training is determined on the training set. The criterion used to determine the best network is the best value of the loss function when training with a loss function and the highest satisfaction ratio if the training is done using the constraints. A LeNet-5 network is used for the experiments on MNIST. A VGG-7 network is used for the experiments on CIFAR10. The constraint based learning (Con) for the CE (1) has been tested for multiple bounds. More specific, it is tested for $(T, F) \in \{(0.45, 0.1), (0.65, 0.1), (0.85, 0.1), (0.75, 0.05), (0.85, 0.05), (0.95, 0.05)\}$. The best results are reported in bolt if they are significantly different according to a t-test with 0.01 $p-$value. The supervised regression task is evaluated on the Bias correction[1] (BC) data set and Family income[2] (FI) data set as described in [3]. The constraint based learning (2) for the MSE is ran for $\delta \in \{0.05, 0.005\}$. The experiments on BC and FI use a multi-layer

[1]Available on `https://archive.ics.uci.edu/ml/datasets/Bias+correction+of+numerical+prediction+model+temperature+forecast`.

[2]Available on `https://www.kaggle.com/grosvenpaul/family-income-and-expenditure`.

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

	T	F	CE	Accuracy	SR
CE	0.45	0.1	0.0512 ± 0.0020	$\mathbf{0.9929 \pm 0.0004}$	0.9377 ± 0.0081
Con	0.45	0.1	$\mathbf{0.0390 \pm 0.0016}$	0.9913 ± 0.0003	$\mathbf{0.9958 \pm 0.0001}$
CE	0.65	0.1	0.0522 ± 0.0030	$\mathbf{0.9925 \pm 0.0003}$	0.9367 ± 0.0072
Con	0.65	0.1	$\mathbf{0.0271 \pm 0.0005}$	0.9912 ± 0.0004	$\mathbf{0.9957 \pm 0.0001}$
CE	0.85	0.1	0.0505 ± 0.0029	$\mathbf{0.9927 \pm 0.0007}$	0.9352 ± 0.0064
Con	0.85	0.1	$\mathbf{0.0259 \pm 0.0008}$	0.9912 ± 0.0004	$\mathbf{0.9958 \pm 0.0001}$
CE	0.75	0.05	0.0524 ± 0.0032	$\mathbf{0.9929 \pm 0.0005}$	0.9325 ± 0.0030
Con	0.75	0.05	$\mathbf{0.0279 \pm 0.0012}$	0.9905 ± 0.0006	$\mathbf{0.9954 \pm 0.0001}$
CE	0.85	0.05	0.0531 ± 0.0042	$\mathbf{0.9929 \pm 0.0003}$	0.9371 ± 0.0070
Con	0.85	0.05	$\mathbf{0.0299 \pm 0.0014}$	0.9902 ± 0.0005	$\mathbf{0.9954 \pm 0.0001}$
CE	0.95	0.05	0.0540 ± 0.0048	$\mathbf{0.9929 \pm 0.0004}$	0.9333 ± 0.0063
Con	0.95	0.05	$\mathbf{0.0349 \pm 0.0018}$	0.9899 ± 0.0006	$\mathbf{0.9953 \pm 0.0001}$

Table 1: The mean and standard deviation of the CE and the SR on the test set of MNIST for CE and constraint based learning (Con).

perceptron [3]. No regularization was used during training as regularization alters the local minima of the loss function and introduces a preference between different solutions that satisfy all the constraints for constraint based learning. The latter is undesirable as this adjusts the space of *optimal* predictions under the constraints and the CSP becomes a constrained optimization problem.

The baseline method for the classification task is obtained by training with CE. The baseline method for the regression task is obtained by training with MSE. The hypothesis is that the proposed method performs equally well or outperforms the baseline method. All methods are trained with 4 pseudo-random seeds for initialization until convergence, resulting in possibly a different number of epochs before convergence for each method and each pseudo-random seed. The initializations are kept the same for the different methods. Next to the loss function, the classification task is evaluated with the Top-1 accuracy and the Satisfaction Ratio (SR), which is the ratio between the satisfied constraints over the total amount of constraints. For a fair comparison, no tuning of the hyperparameters is performed. Therefore, achieving state-of-the-art performance is not the goal of this work.

4 Results & Discussion

First, the results for the classification problem on the test set of the MNIST data set are discussed Table 1. The small differences between CE are a consequence of the usage of different seeds. Even though the difference in Top-1 accuracy is significant, the difference is relatively small for all choices for T and F. The difference increases as a function of T. In terms of CE, there are large differences. It appears that a small T and a large F results in a low CE for the constraints as objective. This is expected as in this case values that are allowed are closer to 0.5

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

	T	F	CE	Accuracy	SR
CE	0.45	0.1	0.3889 ± 0.0107	0.8762 ± 0.0032	0.8352 ± 0.0025
Con	0.45	0.1	0.3898 ± 0.0070	0.8758 ± 0.0029	$\mathbf{0.9725 \pm 0.0003}$
CE	0.65	0.1	0.3897 ± 0.0094	0.8757 ± 0.0027	0.8335 ± 0.0027
Con	0.65	0.1	0.4123 ± 0.0156	0.8761 ± 0.0030	$\mathbf{0.9725 \pm 0.0002}$
CE	0.85	0.1	$\mathbf{0.3898 \pm 0.0051}$	0.8761 ± 0.0018	0.8328 ± 0.0031
Con	0.85	0.1	0.4677 ± 0.0051	0.8780 ± 0.0020	$\mathbf{0.9717 \pm 0.0004}$
CE	0.75	0.05	$\mathbf{0.3894 \pm 0.0059}$	0.8752 ± 0.0018	0.7904 ± 0.0029
Con	0.75	0.05	0.4996 ± 0.0104	0.8770 ± 0.0019	$\mathbf{0.9720 \pm 0.0004}$
CE	0.85	0.05	$\mathbf{0.3895 \pm 0.0100}$	0.8756 ± 0.0007	0.7895 ± 0.0029
Con	0.85	0.05	0.5553 ± 0.0063	0.8757 ± 0.0010	$\mathbf{0.9718 \pm 0.0004}$
CE	0.95	0.05	$\mathbf{0.3900 \pm 0.0079}$	0.8753 ± 0.0016	0.7867 ± 0.0032
Con	0.95	0.05	0.6628 ± 0.0104	0.8739 ± 0.0013	$\mathbf{0.9718 \pm 0.0004}$

Table 2: The mean and standard deviation of the CE and the SR on the test set of CIFAR10 for the CE and constraint based learning (Con).

which results in a small CE. Moreover, when T is large and F is small there is on average and in terms of the standard deviation a significantly smaller CE except for $(T, F) = (0.95, 0.05)$. In other words, large values for T and small F lead to the model being more confident on the test set as the probabilities for the ground truth class label is larger and for all other classes smaller compared to models that are trained using the CE. The SR is consistently higher for constraint based learning, leading to high probabilities when the predicted class is correct.

Second, the results for the classification problem on the test set of the CIFAR10 data set are discussed Table 2. The CE is similar or slightly higher for constraint based learning. In terms of Top-1 accuracy no significant differences can be observed. For the SR there is a large difference between the CE loss and the constraint based learning. This shows that when a correct prediction is made by the neural network that a large probability is assigned to this prediction. Therefore, it might be that when the resulting prediction of the neural network has a relatively low prediction that this is closely related to the example being out-of-distribution compared to the training set.

Third, the results for the regression problem on the test set of the BC data set are discussed Table 3. The results for MSE are the same when the MSE is used as loss function for the different values of δ as the value of δ has no influence on the MSE or the training procedure in this case. The value for SR does change for different values of δ as a larger value for δ results in a constraint that is easier to satisfy as the feasible interval is larger. There are no large differences in terms of MSE and SR when a relatively small value for δ is used. Therefore, it can be concluded that on this data set and for this network architecture the performance is similar for the proposed constraint based learning.

Fourth, the results for the regression problem on the test set of the FI data set are discussed Table 3. The mean and standard deviation of the MSE is lower

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

		BC		FI	
	δ	MSE	SR	MSE	SR
MSE	0.005	0.0020 ± 0.0002	0.5448 ± 0.0063	0.0018 ± 0.0005	0.5779 ± 0.0104
Con	0.005	0.0020 ± 0.0002	0.5538 ± 0.0045	0.0016 ± 0.0004	$\mathbf{0.6774 \pm 0.0244}$
MSE	0.05	0.0020 ± 0.0002	0.8925 ± 0.0054	0.0018 ± 0.0005	0.9418 ± 0.0066
Con	0.05	0.0021 ± 0.0001	0.8929 ± 0.0065	0.0017 ± 0.0004	0.9545 ± 0.0047
MSE	0.1	$\mathbf{0.0020 \pm 0.0002}$	$\mathbf{0.9800 \pm 0.0068}$	$\mathbf{0.0018 \pm 0.0005}$	$\mathbf{0.9854 \pm 0.0037}$
Con	0.1	0.0026 ± 0.0003	0.9771 ± 0.0072	0.0038 ± 0.0002	$\mathbf{0.9905 \pm 0.0024}$

Table 3: The mean and standard deviation of the MSE and the SR on the BC data set and FI data set for the MSE and constraint based learning (Con).

for the constraints when $\delta \in \{0.005, 0.05\}$. The similar observation holds for the mean and standard deviation of the SR for these values of δ with a significantly large difference for $\delta = 0.005$. This illustrates that allowing these small errors during training can lead to a better performance on a test set. In particular, there is a higher probability of achieving a similar error on the test set as the SR is 0.1 bigger for using the constraints with $\delta = 0.005$ compared to using the MSE. Hence, on this data set and for this network we can conclude that the constraints with a small error, that is, a small value for δ, the performance of the resulting model is better.

5 Conclusion & Future work

The proposed CSP reformulated alternatives for the CE loss function and the MSE loss function have each been tested on two independent data sets. It is shown that the constraint based learning performs similarly or better than the traditional loss functions, resulting in a valuable alternative to train networks.

An important direction for future work is to test the constraints on a task where traditional loss functions are known to often overfit. The hypothesis is that the constraints might lead to a lower probability of overfitting. A second direction for future work is to investigate alternatives for other loss functions such as the Kullback-Leibler divergence loss.

References

[1] Kareem Ahmed, Stefano Teso, Kai-Wei Chang, Guy Van den Broeck, and Antonio Vergari. Semantic probabilistic layers for neuro-symbolic learning. In *Proceedings of the 38th Conference on Uncertainty in Artificial Intelligence*, pages 29944–29959. Curran Associates, Inc., 2022.

[2] Kshitij Goyal, Sebastijan Dumancic, and Hendrik Blockeel. Deepsade: Learning neural networks that guarantee domain constraint satisfaction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38:12199–12207, 3 2024.

[3] Quinten Van Baelen and Peter Karsmakers. Constraint guided gradient descent: Training with inequality constraints with applications in regression and semantic segmentation. *Neurocomputing*, 556:126636, 2023.