Neural Criticality: Validation of Convolutional Neural Networks

Václav Diviš, Marek Hrúz
University of West Bohemia

Problem definition

The black-box behavior of Convolutional Neural Networks is one of the biggest obstacles to the development of a standardized validation process. Methods for analyzing and validating neural networks currently rely on approaches and metrics provided by the scientific community without considering functional safety requirements. However, automotive norms, such as ISO26262 and ISO/PAS21448, do require a comprehensive knowledge of the system and of the working environment in which the network will be deployed. In order to gain such a knowledge and mitigate the natural uncertainty of probabilistic models, we focused on investigating the influence of filter weights on the classification confidence in Single Point Of Failure fashion. We laid the theoretical foundation of a method called the Neurons' Criticality Analysis. This method, as described in this article, helps evaluate the criticality of the tested network and choose related plausibility mechanism.

Neural criticality

Firstly, we denote the analyzed convolutional neural network as N, which consists of a set of layers L, containing weights W and biases b and we introduced the criticality metric according to Equation 1

$$f_{cr} = \begin{cases} \hat{y}_i - \hat{y}_{mi}, & \text{if } f_m(x_i) : (\hat{y}_i - \hat{y}_{mi}) \geq \tau \\ \frac{1}{1-\hat{y}_{mj}}, & \text{if } f_m(x_i) : \hat{y}_{mi} \leq \hat{y}_{mj} \text{ and } \hat{y}_{mj} < 0.5 \\ 2, & \text{if } f_m(x_i) : \hat{y}_{mi} \leq \hat{y}_{mj} \text{ and } \hat{y}_{mj} \geq 0.5 \\ 0, & \text{otherwise,} \end{cases} \quad (1)$$

where f_{cr} returns a criticality with domain $[0, 2]$ for a given CNN which is masked, $f_m(x_i) \subset N$. The masking of a CNN is carried out by setting neurons' weights to zero. In case of convolution, all the values of a filter are set to zero. A different kind of error modeling would lead to extensive permutation and was therefore not further investigated.

Neural criticality analysis

Secondly, we constructed the evaluation algorithm 1 which calculates the criticality for a given input image x_i, belonging to class i, drawn from a test set $\mathcal{X}$. We define the task of NCA as the analysis of the neurons' contribution to the classification hypothesis which can be seen as equivalent to the Single Point Of Failure analysis. If all neurons are active, the resulting hypothesis is strong h_{str}, whereas in case a certain amount of neurons have been excluded from the decision, the hypothesis is considered weakened h_{weak}. The neuron's criticality observation of the weakened hypothesis has to be done for every image and class within a test set.

Algorithm 1: NCA algorithm

Data: Let $\mathcal{X}$ be a testing set, i a tested class, N the analyzed CNN, k the number of filters in a layer L and f_{cr} is the criticality function

for *image* $x_i \in \mathcal{X}$ **do**
- $\hat{y}_i = calculate\_conf(N, x_i)$
- $cls_i = predict(N, x_i)$
- **for** *every* L *in* N **do**
  - **for** *every* k *in layer* L **do**
    - $mask\_neuron(k)$
    - $\hat{y}_{mi} = calculate\_conf(N, x_i)$
    - $cls_{mi} = predict(N, x_i)$
    - $criticality = f_{cr}(\hat{y}_i, \hat{y}_{mi}, cls_i, cls_{mi})$

Experiment setup

The motivation behind testing different network architectures was to see the influence of models' chronological improvements on the decision stability, such as residual connections, depthwise convolution and scaling. We therefore evaluated VGG16, Resnet50V2, MobileNetV2 and EfficientNetB0, all pre-trained Keras models on ImageNet. We chose two classes, "street sign" and "mountain bike", in order to evaluate the criticality. For each class, 150 samples were taken. All samples had ground-truth confidence higher than 0.8 so that we ensured that kernels' responses would be highly excitated. Adversary samples were generated by non-target FSGD method until either achieving a confidence greater than 0.5 or ending after 20 iterations. For all tests we set the criticality threshold τ to 0.0, which allows the algorithm to measure and visualize the criticality of all neurons and distinguish between critical and anti-critical ones. In practice, the threshold should be justifiable via hazard and risk assessment and will be presumably higher than 0.0.

Results

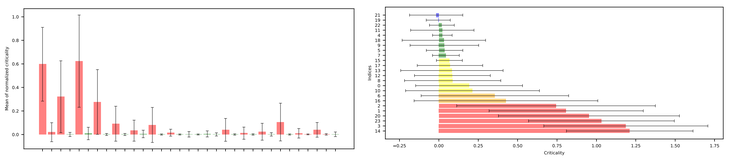

Fig. 1: Analysis of MobileNetv2 architecture showed a higher instability caused by projection layers. The image on the left shows weighted criticality per layer, whereas the image on the right depicts the 20 most critical neurons of the block_1_project layer.

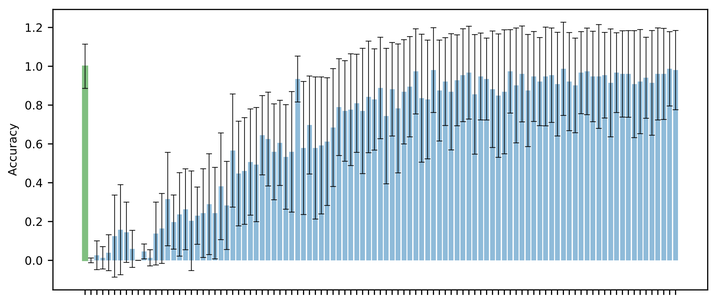

Fig. 2: Accuracy stability of the 100 most critical neurons on normal dataset (for class "mountain bike"), showing a gradual increase of accuracy with respect to decreasing neurons' criticality in case of MobileNetv2 model.

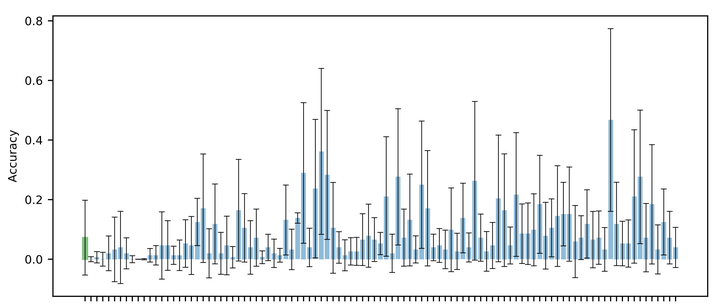

Fig. 3: MobileNetV2's accuracy stability of the 100 most critical neurons, taken from the analysis of a normal dataset and evaluated on an adversary dataset. Neurons with increased accuracy could be further used for diagnosis purposes, but have to be chosen with respect to both mean and standard deviation of the resulting accuracy. In such a diagnostic case, masking multiple neurons could be desirable and would lead to higher diagnoses accuracy.

Conclusion

We dedicated a great part of our work to introducing and testing an innovative method, the Neurons' Criticality Analysis. The outcome of this analysis was a comprehensive report diagram depicting the criticality of each layer and each neuron of evaluated model. The domain of criticality is $[-1, 2]$. We discussed that masking neurons with negative criticality can also have a positive influence on the model's decision confidence. We called this behavior "anti-critical". The inter-class anti-critical neurons could hypothetically be removed from the decision process. This idea led us to the conclusion that the correlation between the neurons removed during the pruning process and the anti-critical neurons discovered via NCA should be further investigated.

We claimed that using spatial aggregation via projection layers may on the one hand improve the high dimensional feature representation [1], but on the other hand creates very critical dense connections, especially in the shallow layers, as we pointed out. From functional safety point of view this isn't necessarily negative, since the plausibility function could be applied to only a concentrated area of neurons. In addition, some critical neurons showed the ability to increase mean accuracy on adversary dataset, which could be used in order to discover adversary attacks and irregularities during inference. We hypothesize that an equilibrium between the position of the first projection layer, number of critical neurons and models' accuracy should be further investigated.

As aforementioned, the purpose of NCA is to identify critical neurons. With further measures, the mean and standard deviation of the criticality should be decreased and the flawless calculation of the neuron should be diagnosed. Concretely this can be achieved by several approaches, such as:

- fine-tuning of the model with deterministic dropout and loss which will incorporate the layers criticality
- plausibility check of the critical neurons or layers or redundant computational branch results
- storage of the neurons' weights and biases in two places in RAM and comparing them
- introduction of inverse layers in order to compute and evaluate the original inputs over critical connections

Our method can also be used for Out-of-Distribution detection, where instead of randomly sampling sub-networks predictions, as it is done by MC dropout, deterministic dropout would be based on several highly critical neurons for every class. Such an approach would decrease the computational demand and arguably increase the reliability and transparency of such a network. In order to encourage additional experiments and deeper explorations, we published our code and supplement results on GitLab

`https://gitlab.com/divisvaclav/cnn_eval_tool/-/tree/wo_gui_branch`

Acknowledgements

The work has been supported by the grant of the University of West Bohemia, project No. SGS-2019-027.

References

[1] Christian Szegedy et al. "Rethinking the inception architecture for computer vision". In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2016, pp. 2818–2826.