Deep Ensemble Confidence Levels for Multi-hot Categorization

Giovani Leone[a]

[a]*The University of North Carolina at Chapel Hill*

Introduction

NOICE (Neural Optical Image Categorizer for the E-log) is a small collaboration tasked with categorizing the images in the Fermilab Accelerator Division electronic logbook by using Artificial Intelligence. To do so, we manually categorized a subset of the images in the E-log into nine independent labels. Each image was then multi-hot-encoded into a nine-dimensional binary vector because each image could have more then one label.

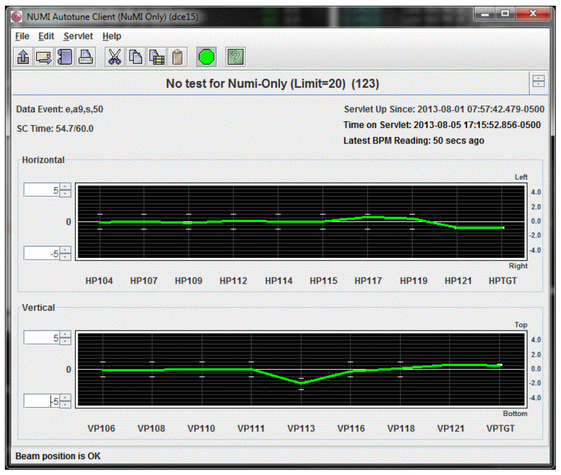

Figure 1. An example of an image that is both a Plot and an Application (possible labels: "Application", "Parameter Page", "Plot", "Document", "Drawing", "Photograph", "Diagram", "NOICE", "Undefined"). Thus, the multi-hot-encoding for this image is [1,0,1,0,0,0,0,0,0].

A Deep Ensemble is an ensemble of deep neural networks[1]. This Deep Ensemble is composed of 100 different random initializations of a deep neural network. This deep neural network was built in TensorFlow2 (version 2.3) and the architecture is described in Figure 2. A prediction with the Deep Ensemble thus gives a distribution of 100 output sigmoid activation scores for each label of each image. The scores on a given label were then compiled to determine the ensemble's verdict. A confidence level was then assigned to each label of each image by using the output sigmoid activation scores as a measure of a model's *self-confidence*. The collection of the models' *self-confidences* were then used to generate an ensemble confidence level.

1. 2D Convolution, 16 filters, 3x3 kernel, ReLU Activation
2. 2D Max Pooling, 3x3 pool size
3. Dropout, 0.1 dropout rate
4. 2D Convolution, 32 filters, 3x3 kernel, ReLU Activation
5. 2D Max Pooling, 3x3 pool size
6. Dropout, 0.1 dropout rate
7. Flatten
8. Dense, Sigmoid Activation

   Loss: Binary Cross-Entropy

Figure 2. The deep neural network architecture that was used to form the Deep Ensemble. Note that the output activation function is a sigmoid function.

Ensemble Output Distribution

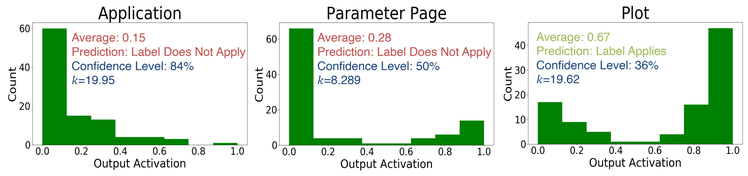

Figure 3. Output sigmoid activation scores distribution of the application, parameter page, and plot labels of a random test set image. The average of the distributions are 0.15 for the "Application" label, 0.28 for the "Parameter Page" label, and 0.67 for the "Plot" label. Following the procedure outline in Confidence Level Calculation, the confidence levels of the distributions are 84% for the "Application" label, 50% for the "Parameter Page" label, and 36% for the "Plot" label.

Since the output sigmoid activation scores treat each label independently, I defined the ensemble to guess that a label applies to an image if and only if the average of the output sigmoid activation scores is greater than 0.5. As shown in Figure 3, the distribution of output sigmoid activation scores can vary in spread and modality.

Confidence Level Calculation

Treating the sigmoid output activation score on a label of an image as a measure of a model's confidence in that image, we assumed the following: a score of 0 corresponds to 100% confidence that the label does not apply, a score of 0.5 corresponds to 0% confidence that the label applies and does not apply, and a score of 1 corresponds to 100% confidence that the label does apply. Thus, we defined the confidence level made by an N model ensemble on a choice of labeling as

$$\boldsymbol{C} = \left| \sum_{n=1}^{N} \boldsymbol{c}\big(\boldsymbol{s}(\boldsymbol{n})\big) \right|$$

with the normalization condition of $\mathbf{1} = \left|\sum_{n=1}^{N} \boldsymbol{c}(\mathbf{1})\right|$, where $c(s(n))$ is the *self-confidence functional*, and $s(n)$ is the output sigmoid activation score of the n^{th} model.

The *self-confidence functional* $c(s(n))$ can be chosen such that a calibration condition is met. A sigmoid-shaped *self-confidence functional,* centered at $s(n) = 0.5$,

$$c_{sigmoid}\big(s(n)\big) = A\left(\frac{e^{k(s(n)-0.5)}}{e^{k(s(n)-0.5)}+1} - 0.5\right)$$

was used. A was determined by the normalization condition and a k was chosen for each label such that the accuracy of a label over all images equals the average of the confidence levels on that label over all images (the calibration condition). Note in Figure 4 that as k approaches 0, $c_{sigmoid}\big(s(n)\big)$ becomes symmetric about $s(n) = 0.5$, and that as k approaches infinity, $c_{sigmoid}\big(s(n)\big)$ becomes a signum function, scaled by A.

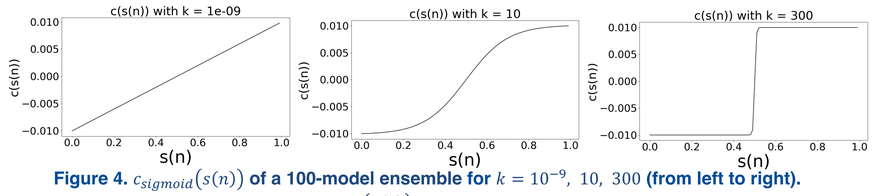

Figure 4. $c_{sigmoid}\big(s(n)\big)$ of a 100-model ensemble for $k = 10^{-9},\ 10,\ 300$ (from left to right). Note that as k approaches 0, $c_{sigmoid}\big(s(n)\big)$ becomes symmetric about $s(n) = 0.5$, and that as k becomes large, $c_{sigmoid}\big(s(n)\big)$ approaches a signum function, scaled by A.

Conclusions

Using the procedure described in **Confidence Level Calculation**, the confidence levels for the distributions in Figure 3 are 84% for the "Application" label, 50% for the "Parameter Page" label, and 36% for the "Plot" label. By following the calibration procedure described in **Confidence Level Calculation**, the k values of $c_{sigmoid}$ are 19.95 for the "Application" label, 8.289 for the "Parameter Page" label, and 19.62 for the "Plot" label. Thus, by design of the sigmoid-shaped *self-confidence functional*, a wider spread or bimodality yields a lower confidence level, with the lowest calculable confidence level being 0 for a uniform distribution (by the symmetry of the sigmoid-shaped *self-confidence functional*).

A *self-confidence functional* can be calibrated to a deep ensemble's accuracy and used to calculate the confidence levels on labels for a multi-hot-encoded Deep Ensemble (by extension, this is also applicable to single-hot-encoded models utilizing sigmoid output activation functions). Future explorations of this technique include evaluating the confidence levels of labels across a large data set and comparing the average confidence level calculation of a label to the Deep Ensemble's accuracy on a large, unseen data set.

References

[1] B. Lakshminarayanan, A. Pritzel, and C. Blundell, ArXiv:1612.01474 [Cs, Stat] (2017).

Acknowledgements

I would like to thank my mentor, Jason St. John, for his guidance and support, Kyle Hazelwood for his knowledge of TensorFlow2 and the code that he contributed to NOICE, Justin Rower and Terence Njekeu for their contributions to the categorization of the dataset which was used throughout this project, and Fermilab for hosting this. This work was supported in part by the U.S. Department of Energy, Office of Science, Office of Workforce Development for Teachers and Scientists (WDTS) under the Science Undergraduate Laboratory Internships Program (SULI).