Information of the last hidden-layer in neural network: Represetation and Application on Calibration

Xin-Yu Hu, and Luo-Ye Zou

Abstract: Outputs of hidden layer nodes in a neural network are usually regarded as features extracted from the input, sharing significant information with the output. We focus on the last hidden layer with a sigmoid activation function whose outputs are concentrated, and encode each node by using binomial distributions and Gaussian mixture models. We then calculate the mutual information between each node and the final output, demonstrating the effectiveness of feature representations with numerical experiments. Finally, we apply the encoding for confidence estimation, exploring its calibration potential.

1 Introduction

In neuroscience, encoding the output of neurons under different stimuli is a common approach. By analyzing the distributions from encoding, one neuron's response to various stimuli can be estimated, thereby evaluating its representation for a particular task. Related work shows mutual information can be used to measure the association of neurons and different tasks. In neural networks, outputs of hidden layer nodes are usually regarded as features extracted from the input. Task Variance [1] is common to analyzing hidden layers, which do not involve output information. However, according to the Information Bottleneck theory [2], we suppose that in a well-trained neural network, each nodes in the last hidden layer should share significant information with the final output as well. In this work, we focus on the last hidden layer with a sigmoid activation function and encode its nodes using binomial distributions and Gaussian mixture models. Numerical experiments indicate a significant positive correlation between mutual information and task variance, demonstrating the effectiveness of the feature representations. We also apply the encoding for calibration, yielding promising results.

2 Information Representation of the Last Hidden Layer

2.1 Encoding

Sigmoid is a commonly used activation function, and its output tends to saturate when the input is very large. As a result, outputs after sigmoid activation are often concentrated during feedforward. We consider two neural networks: a simple two-hidden-layer network and VGG16, processing the MNIST and CIFAR-100 datasets respectively. The trained model performs the test task on the training set, and when it gives the correct result, we record the last hidden layer outputs.
In the simple two-hidden-layer neural network, the width of the last hidden layer is 32, with an output of 10 categories, and the test accuracy reaches 97%. We observe that the majority of the node outputs are concentrated around 0 and 1. Specifically, most of the outputs fall within the range of 0.9 to 1 and 0 to 0.1, while the rest accounts for approximately only 3%. Therefore, we encode the last hidden layer outputs with a binomial distribution, where nodes with outputs less than 0.5 are encoded as 0, and those greater than 0.5 are encoded as 1. The second neural network, VGG16, is a classic convolutional neural network with a last hidden layer width of 512 and an output of 100 categories, achieving a test accuracy of 74%. The hidden layer outputs of VGG16 are not concentrated around 0 and 1, so we use a Gaussian mixture model to encode outputs in the hidden layer and test its R-square error. When encoding the nodes separately across different outputs, the average fitting performance of three-component Gaussian mixture model reached 95.2%. When encoding the nodes together across different outputs, the average fitting performance of ten-component Gaussian mixture model reached 81.1%. Therefore, we used the Gaussian mixture model to encode the last hidden layer outputs.

2.2 Mutual Information, Task Variance, and Bias

To estimate the representation of each node in the last hidden layer, we calculate the mutual information between each node and the final output: $I(h_i, y) = H(h_i) + H(y) - H(h_i, y)$, where h_i denotes the i-th hidden layer node, y denotes label, and H denotes entropy. For the Gaussian mixture model, $H(y) = \int_y p(y) \log p(y)\, dy = \int_y \left(\sum_w w p_w(y)\right) \log \left(\sum_w w p_w(y)\right)\, dy$ is difficult to compute directly. Noting that

$$\left(\sum_w w p_w(y)\right) \log \left(\sum_w w p_w(y)\right) \geq \left(\sum_w w p_w(y)\right) \sum_w w \log p_w(y) = \sum_{w_1, w_2} w_1 w_2 p_{w_1}(y) \log p_{w_2}(y),$$

where w is the weight of Gaussian model and $p_w(y)$ is the corresponding probability density function, we use this lower bound to approximate the calculation. This leads to

$$H(y) \approx \sum_{w_1, w_2} w_1 w_2 \left(\frac{(\mu_1 - \mu_2)^2 + \sigma_1^2}{2\sigma_2^2} + \frac{1}{2} \log \sigma_2 + \log 2\pi \right),$$

where μ represents the mean of Gaussian model and σ represents the standard deviation. $H(h_i, y)$ is decomposed as $H(y|h_i) + H(h_i)$, and then same approximation method is applied.

Task variance [1] (TV) effectively represents the activity of neural network nodes, defined as the variation in the hidden layer outputs across different data points: $Var[h_i] = E[h_i^2] - E[h_i]^2$. Bias are the internal parameters of the neural network model. The transformation in the output layer is written as $f(h) = Wh + b$, where W is the weight matrix and b is the bias vector. To obtain the effect of the bias on the hidden layer, we rewrite the function as $f(h) = W(h + b')$, such that the bias directly affects the hidden layer. When solving the indeterminate equation $b = Wb'$, we assume that the same bias affects each node in the same way.

With the results from Section 2.1, we computed the Pearson correlation coefficient between the three metrics. As shown in Table 1, a significant positive correlation is observed between mutual information and task variance in both networks. This suggests that in the last hidden layer of a well-trained neural network, nodes with larger variations share more information with the output, demonstrating the pattern that active nodes do contain more feature information. In the two-hidden-layer neural network, the bias also exhibits a strong positive correlation with mutual information and task variance. Since the level of the bias can be seen as a form of Incentive or inhibition for nodes [3], it implies the neural network model has learned the pattern. However, in VGG16, only a weak positive correlation is observed. We suppose that this is because VGG16's performance on CIFAR-100 is not as effective as the simple neural network's performance on MNIST.

ρ	MI	TV	Bias
MI	1	0.949	0.493
TV	0.949	1	0.472
Bias	0.493	0.472	1

ρ	MI	TV	Bias
MI	1	0.674	0.057
TV	0.674	1	0.099
Bias	0.057	0.099	1

Table 1: Pearson correlation coefficients between Mutual Information (MI), Task Variance (TV), and Bias for each node in the hidden layer. Left: Results for the MNIST task. Right: Results for the CIFAR-100 task.

3 Calibration Test

We use the calibration test as an example to demonstrate the representational capacity of the final hidden layer information. Calibration test refers to aligning a classifier's output probability to the true probability. Previous studies have shown that the softmax outputs of neural networks tend to overestimate prediction probabilities, i.e., they are overconfident. We hypothesize that in a well-trained neural network, when presented with training-data-like inputs, the hidden layer outputs exhibit a certain pattern, while for other inputs, they will differ. In other words, the hidden layer contains information that can represent confidence. Thus, we use the encoding results to estimate prediction confidence for test data. Specifically, the distribution of the training data which the neural network has successfully learned is denoted as A. It is assumed that a new data point x belongs to this set, its output label denoted as y. We compute the log-likelihood of the hidden layer output as

$$L(h|x \in A, y = l) = \sum_i L(h_i|x \in A, y = l),$$

which is the prediction confidence for the new data.

To better fit the hidden layer output patterns, we improve the distributions used for encoding. Three schemes are considered: trinomial distribution (with three categories: less than 0.1, greater than 0.9, and others), Beta distribution $B(\alpha, 1-\alpha)$ (where α is the parameter), and the Gaussian mixture model. We apply the first two schemes to the two-hidden-layer neural network, while the Gaussian mixture model is for the VGG16 network. We use Area Under the Receiver Operating Curve (AUROC) to evaluate the calibration performance. The experiment is performed on simple two-hidden-layer neural network and VGG16, with the testing sets from MNIST and CIFAR-100, respectively. The baseline [4] is the confidence directly from the Softmax output. As shown in Table 2, our method achieves calibration performance all superior to the baseline on both datasets.

AUROC	Softmax	Trinomial	Beta	GMM
MNIST	0.953	**0.960**	0.958	/
CIFAR100	0.864	/	/	**0.873**

Table 2: AUROC for calibration on different datasets. A higher AUROC value indicates better calibration performance.

Contribution
Xin-Yu Hu: Encoding, computation of metrics and calibration. Paper writing.
Luo-Ye Zou: Training neural networks. Data Record. Paper Tyeseting.

References

[1] G. R. Yang, M. R. Joglekar, H. F. Song, W. T. Newsome, and X.-J. Wang, "Task representations in neural networks trained to perform many cognitive tasks," *Nature Neuroscience*, vol. 22, pp. 297 – 306, 2019.

[2] N. Tishby and N. Zaslavsky, "Deep learning and the information bottleneck principle," *2015 IEEE Information Theory Workshop (ITW)*, pp. 1–5, 2015.

[3] E. Williams, A. H.-W. Ryoo, T. Jiralerspong, A. Payeur, M. G. Perich, L. Mazzucato, and G. Lajoie, "Expressivity of neural networks with random weights and learned biases," *ArXiv*, 2024.

[4] J. Gawlikowski, C. R. N. Tassi, M. Ali, J. Lee, M. Humt, J. Feng, A. M. Kruspe, R. Triebel, P. Jung, R. Roscher, M. Shahzad, W. Yang, R. Bamler, and X. Zhu, "A survey of uncertainty in deep neural networks," *Artificial Intelligence Review*, vol. 56, pp. 1513–1589, 2021.