An Uncertainty Measure based on Logits in Deep Learning Classification Tasks

Huiyu Wu huiyu.wu1@northwestern.edu
Diego Klabjan d-klabjan@northwestern.edu

Abstract: We introduce a new, reliable, and agnostic uncertainty measure for deep learning classification tasks called *logit uncertainty*. This is based on the logit outputs of any neural network and can be extended to include some other classifiers. We then show that this new uncertainty measure yields a superior performance compared to existing uncertainty measures on different tasks, including out of sample detection and finding wrong predictions. We explore the relationship between our method and high-density regions. We also demonstrate a new approach to test uncertainty measures using intermedia outputs in training of generative adversarial nets.

1 Introduction

Machine learning has seen drastic accuracy improvements in classification tasks over the past few years with ever increasing computational power and deeper Neural Nets. For example, Top-1 Accuracy for ImageNet can exceed 85% with a state-of-the-art method [1]. Despite the incredible accuracy achievements, Neural Net classifiers inevitably make mistakes, some of which can be costly. Therefore, it is beneficial to know the uncertainty associated with the classification output of the neural nets so that we know when our models are more likely to make mistakes.

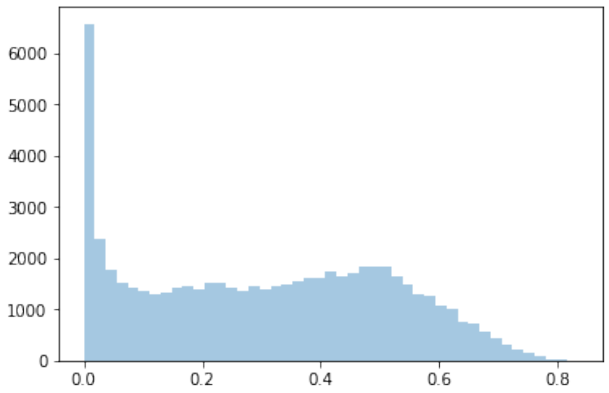


Figure 1. Uncertainty on incorrect predictions using Ensemble

Many methods of evaluating uncertainty of classifications rely upon the softmax probabilities generated from the Neural Nets. However, these probabilities or the entropies of these probabilities are notoriously unreliable, making these uncertainty measures unsuitable for tasks such as medical diagnosis or fraud detection when mistakes can be costly. For example, Figure 1 shows the uncertainty histogram on a data set that is completely different from the training set obtained using an ensemble method and it shows the high portion of low uncertainty predictions. In this paper, we derive a reliable uncertainty measure based on the logit outputs of neural nets named logit uncertainty. The measure can help classifiers detect when they are more likely to make mistakes. The usage of such uncertainty measure may include, for

example, introducing an expert into the decision-making process when the uncertainty associated with the classification is high. Another application of such an uncertainty measure is in novelty detection, where the neural net can detect a shift in the data distribution [8]. Thus, we can make the decision to retrain the classifier to adapt to the shift.

From empirical experiments, we believe the logit outputs capture the data uncertainty, meaning if class A is intrinsically similar to class B but different than class C, then the logit value at class B of class A's logit output is higher than the logit value at class C of the same logit output. For example, in the Cifar10 image classification task, we observed that logit values at cat for dog images' logit outputs are higher compared to the logit values at other classes such as trucks or airplanes and vice versa (Figure 2). Our innovation is to use the Gaussian Mixture to model the logit outputs of correctly predicted training data for each class and to compute uncertainty values based on the probability density function of the Gaussian Mixtures.

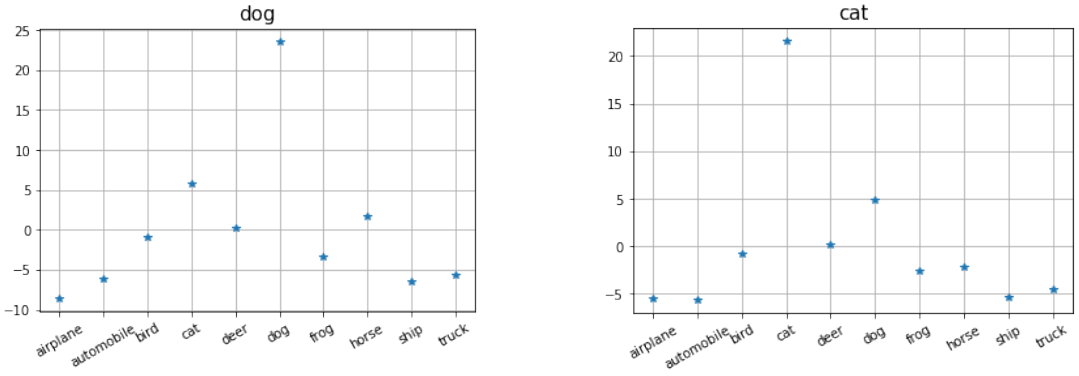


Figure 2. Average logit values of correctly predicted Dog and Cat images

The logit uncertainty we derive is agnostic and only depends on the raw logit output of the neural net. Therefore, we can treat the neural net classifier as a black box and eliminate the need to retrain the model in order to compute the uncertainty value. This is especially beneficial since the training of a complex model on a challenging task can take days or even weeks to complete. As a result, our method to compute uncertainty is easy to incorporate into existing deep learning models in serving.

In our experiments, we demonstrate that our logit uncertainty outperforms existing uncertainty measures by a large margin. For example, unlike other methods, our uncertainty measure exhibits a clear distinction for images of airplanes, trucks, and cars tested on neural net trained on passenger cars. Additionally, the order relationship of the uncertainty value obtained from our method is meaningful, with uncertainty values for airplanes > trucks > cars for the neural net trained on cars. This is another desired quality that many existing methods do not exhibit [2].

Our main contributions are as follows; We propose a new uncertainty measure relying on logit values. The measure does not rely on the underlying classification model. We also analyze the relationship between our uncertainty measure to high density regions. The third contribution is embedded in the comprehensive experimental study. To this end, we introduce evaluation consisting of training on one data set and assessing uncertainty on a different data set with context drift. We also experiment by generating GAN [7] images based on a data set and by then computing uncertainty on such samples. A comparison of our logit-based uncertainty measure to other prior suggestions shows that our strategy is much more reliable.

This paper is organized as follows; Section 2 explores existing uncertainty measures and each methods' weakness. Section 3 introduces and analyze our method. Section 4 describes our comprehensive experiments. Finally, Section 5 presents our conclusions.

2 Literature Review

Our line of work is related to uncertainty and confidence estimation for deep learning classifications, which aims to generate a number typically within 0 and 1 from outputs of neural nets. Sensoy et al. provides a novel idea of viewing the logit outputs of neural nets as evidence for each class, and build a Dirichlet distribution with parameters based on those evidence [5]. The authors cleverly tailored a loss function that suits their need and was able to generate meaningful uncertainty measures. However, in practice, their method requires retraining of the neural net which can be expensive. Additionally, their uncertainty values tend to concentrate on 1 when encountering novel data points, making the order relationship of the uncertainty values less meaningful. Our method addresses both of the problems and achieved stronger results in more general tests.

Many other existing methods that compute uncertainty require a number of outputs for a single data point and these methods typically generate the uncertainty values base on the averages or the variance of the outputs. These include the ensemble method [3], the drop-out method [4], and the Bayesian neural nets [6]. Similar to our uncertainty values, their values do not concentrate on a single value compared to [5] and have some evidence of a meaningful order relationship with higher uncertainty actually representing lower confidence. However, from our experiments, when encountering out of the distribution data points, these methods exhibit poor performance compared to our method. Additionally, since all of the above require multiple outputs for a single data point, they all require additional computational power, which is not practical.

One agnostic method to compute uncertainty is provided in [2]. The authors propose to first construct a α-high-density-set of each class and then compute the uncertainty based on the distances of the given points to the constructed sets. Similar to our method, this uncertainty measure is agnostic and only requires minimal computational resources to generate the uncertainty value. However, this method does not work well on high-dimensional data sets such as MNIST, Cifar10, and Cifar100. Our method does not construct density sets of each class, but models the logit outputs using mixtures directly and empirically obtains much better uncertainty estimates on those high-dimensional data sets.

3 Our Method

We have observed that logit values captured some uncertainties, meaning that the logit outputs of a specific class should share similar logit values in each dimension of the logit vector. The intuitive explanation for the above logic is that one class should share similar features. For example, dogs are more similar to cats than to airplanes and therefore all dog images' logit outputs should all have logit values at cats' dimension higher than at airplanes' dimension. With such intuition in mind, we worked with the raw logit outputs of the correctly predicted training data. We fit Gaussian Mixture models to the logit vectors of correct classifications for each class because the GMMs captures the multimodal feature we observed in the logit vectors. One natural idea that follows is to base the uncertainty measure on the density functions of the GMMs, meaning the larger the density function value the smaller the

uncertainty. To work with the extreme small values of the density functions that are typically observed in experiments, we design a score function to make the values more manageable. Lastly, we use a sigmoid function to map the values obtained from the score function on a range between 0 and 1.

3.1 Uncertainty Measure

Suppose we have a data set with k classes and we have already trained a classifier on this dataset. For each class i, we select the correctly predicted training data and use a Gaussian Mixture to model the logit outputs of these data. The number of components is selected by plotting the Bayesian Information Criterion against the number of components and use the 'elbow rule' to select the optimal number of components. Suppose the fitted Gaussian Mixture has probability density function $gmm_i(X)$. We then define a score for a random vector X from $gmm_i(X)$:

$$s_i(X) = \ln(max_t\ (gmm_i(t)) - \ln\big(gmm_i(X)\big)$$

The score s_i is then also a random vector, and that we can find the q-th percentile of the score denote as s_{iq}.To have the final uncertainty value between 0 and 1, we use a logistic function to map the score to uncertainty:

$$g_i(s) = \frac{1}{1 + e^{-m(s - x_0)}}$$

In order to find the parameters m and x_0 of the logistic function, we impose two assumptions:

1. Data samples with scores s_{iq_1} have uncertainty value of u_1, meaning $g_i\big(s_{iq_1}\big) = u_1$.
2. Data samples with scores s_{iq_2} have uncertainty value of u_2, meaning $g_i\big(s_{iq_2}\big) = u_2$.

With the above two assumptions, we have the following system of equations:

$$\frac{1}{1 + e^{-m(s_{iq_1} - x_0)}} = u_1$$

$$\frac{1}{1 + e^{-m(s_{iq_2} - x_0)}} = u_2$$

with solutions:

$$x_0 = \frac{s_{iq_2}\ln(u_1^{-1} - 1) - s_{iq_1}\ln\,(u_2^{-1} - 1)}{\ln(u_1^{-1} - 1) - \ln\,(u_2^{-1} - 1)}$$

$$m = \frac{-\ln\,(u_2^{-1} - 1)}{s_{iq_2} - x_0}$$

Then when we encounter a new data sample x_1 that is classified as class i. The uncertainty value for this new sample will be: $u(x_1) = g_i(s_i(x_1))$.

3.2 Analysis

In this section, we analyze the proposed logit uncertainty to make sure it makes intuitive sense. Additionally, we explore the relationship of our uncertainty measure with existing statistical concepts. The first result follows from monotonicity.

Claim 1: For $x_1$and $x_2$that are predicted as class i, if $gmm_i(x_1) > gmm_i(x_2)$, then we have $u(x_1) < u(x_2)$.

The first claim ensures our logit uncertainty makes intuitive sense. The uncertainty we are trying to compute is similar to the problem of estimating the probability that a new data point in the k-dimensional Euclidean space belongs to the Gaussian mixture for the predicted class. It makes sense that points with higher density values should have lower uncertainties.

Before moving onto the second claim, we introduce highest density region (HDR) [10]. The $(1-\alpha)$-HDR is the subset $R(f_\alpha)$ of the sample space X such that:

$$R(f_\alpha) = \{x: f(x) \geq f_\alpha\}$$

Where f_α is the largest constant such that $\mathrm{P}\big(\mathrm{X} \in R(f_\alpha)\big) \geq 1-\alpha$, and $f(x)$ is the probability density function of x. We have the following result that builds a connection between our logit uncertainty and HDR.

Claim 2: Any sample x within the q_1-HDR have uncertainty value $u(x) < u_1$. Similarly, any sample x within the q_2-HDR have uncertainty $u(x) < u_2$.

Proof: The q_1-HDR of the Gaussian mixture for class i, is the subset $R(f_{1-q_1})$ of the sample space of X such that

$$R\big(f_{1-q_1}\big) = \{x: gmm_i(x) \geq f_{1-q_1}\}$$

Where f_{q_1} is the largest constant such that $\mathrm{P}\left(\mathrm{X} \in R\big(f_{1-q_1}\big)\right) \geq q_1$.

Therefore, we have:

$$\forall x, \mathrm{P}\big(\, gmm_i(x) < f_{1-q_1}\,\big) < 1-q_1$$
$$\mathrm{P}\big(\ln(\, gmm_i(x)) < \ln(f_{1-q_1})\,\big) < 1-q_1$$
$$\mathrm{P}\big(s_i(x) < \ln(max_t\,(gmm_i(t))\,/\,f_{1-q_1})\big) \geq q_1$$

Therefore,

$$\ln(max_t\,(gmm_i(t))\,/\,f_{1-q_1}) = s_{iq_1}.$$

Since

$$\forall x \in R\big(f_{1-q_1}\big), \qquad gmm_i(x) \geq f_{1-q_1}$$
$$s_i(x) < \ln(max_t\,(gmm_i(t))\,/\,f_{1-q_1})$$

Then

$$g_i(s_i(x)) < g_i(\ln(max_t\,(gmm_i(t))\,/\,f_{1-q_1}))$$
$$g_i(s_i(x)) < g_i(s_{iq_1})$$
$$u(x) < u_1$$

We can similarly proof the statement for u_2.

■

The second claim builds a connection between our proposed method of logit uncertainty and HDR. The claim also enables us to adjust the parameters of our logit uncertainty based on different needs. For example, if we are detecting tumors and misclassifications are costly, we can adjust our model to be more conservative with lower values of $q_1, u_1,\ q_2, u_2$.

Another result follows from the second claim. Here is the definition for confidence value $\kappa \in [0,1]$ and confidence region $\mathcal{R} \subseteq \Omega$ for a probability density function $0 \leq p(x) < \infty, \forall x \in \Omega$. κ is a confidence value related to a non-unique region $\mathcal{R}$ such that:

$$\int_{\Omega \setminus \mathcal{R}} p(x)dx = \kappa$$

We should note that the confidence value is not useful unless if we define $\mathcal{R}$ is defined as a minimal volume region that satisfy the previous integral. The resulting confidence region $\mathcal{R}$ with confidence value κ then becomes the HDR $R(f_\kappa)$ [10]. Therefore, we can rephrase claim 2 as any sample x within the $(1-q_1)$ confidence region has uncertainty value $u(x) < u_1$ and any sample x within the $(1-q_2)$ confidence region has uncertainty value $u(x) < u_2$.

For Gaussian mixtures with one component, the multivariate Gaussian case, we explore the relationship between the logit uncertainty and the Mahanobis distance to Gaussian distributions. The Mahanobis distance $r(X)$ between X from a Gaussian and the same Gaussian distribution, with the Gaussian distribution have mean μ and covariance matrix Σ is defined as $r(X) = ((X-\mu)^T\Sigma^{-1}(X-\mu))^{1/2}$. Furthermore, we can define $F(r)$ as the cumulative distribution function of the random vector r.The following result builds a connection between HDR and r.

Claim 3: For $(1-\alpha)$-HDR, $R(f_\alpha)$, of a multivariate Gaussian distribution with mean μ and covariance matrix Σ, and probability density function ϕ. If we let $r_\alpha = F^{-1}(1-\alpha)$, then for any $x \in R(f_\alpha)$, the Mahanobis distance with respect to the Gaussian distribution $r(x) < r_\alpha$.

Proof:

$$\forall x, \mathrm{P}\big(x \in R(f_\alpha)\big) \geq 1-\alpha$$
$$\mathrm{P}(\phi(x) \geq f_\alpha) \geq 1-\alpha$$

Using the probability density function of a multivariate Gaussian, we have:

$$\mathrm{P}(r^2(x) < 2 * \ln(f_\alpha * c)) \geq 1-\alpha$$

Where $c = 2\pi^{k/2}\det(\Sigma)^{1/2}$, k is the dimensionality of the Gaussian,

$$\mathrm{P}\big(r(x) < (2 * \ln(f_\alpha * c))^{1/2}\big) \geq 1-\alpha$$

By definition of cumulative distribution functions,

$$r_\alpha = F^{-1}(1-\alpha) = (2 * \ln(f_\alpha * c))^{1/2}$$

For $\forall x \in R(f_\alpha)$ we have:

$$\phi(x) \geq f_\alpha$$
$$r(x) < (2 * \ln(f_\alpha * c))^{1/2}$$
$$r(x) < r_\alpha$$

■

Combining claim 2 and 3, we can argue that when the Gaussian Mixture has only one component, samples that share the same logit uncertainty have the same Mahanobis distance to the Gaussian distribution.

3.3 Extensions

Our method also applies to logistic regression, support vector machines, and gradient boosting machines. For logistic regression, it can be viewed as a single layer neural network therefore our logit uncertainty applies. SVMs use hyperplane to separate different classes and will output the distance to the hyperplane given a data point. To obtain our logit uncertainty, one can fit GMMs to the distances obtained from the training data set. Therefore, our method also applies to SVMs. Gradient boosting machine outputs logits for each class when used for classification which makes it easy to compute our logit uncertainty. However, our method is not applicable to k-nearest neighbors, random forests, or decision trees. These methods, when used for classification, do not output values that can be fitted using a GMM, therefore, our method does not apply to these classifiers.

4 Experiments

In this section we empirically evaluate the results of our proposed approach, assessing the performance of our logit uncertainty on different tasks described below. In all of the below tests, we used $q_1 = 80$, and $q_2 = 60$, with $u_1 = 0.5$, and $u_2 = 0.2$.

4.1 MNIST data set tests

MNIST is a data set of handwritten images with 60,000 training examples and 10,000 testing examples, where each image is of size 28 × 28. For the training of MNIST, we used CNN with 20 and 50 filters with size 5 × 5 at the first and second convolutional layers and 500 hidden units for the fully connected layer. Our first test involves inspecting the uncertainty distribution on the incorrectly classified images in the test set, Figure 3 shows the distribution of uncertainty values for such images. As a comparison, we have also included distribution for the incorrectly classified images using ensemble method detailed in [3] and Bayesian approximation in [4], and evidential method in [5].

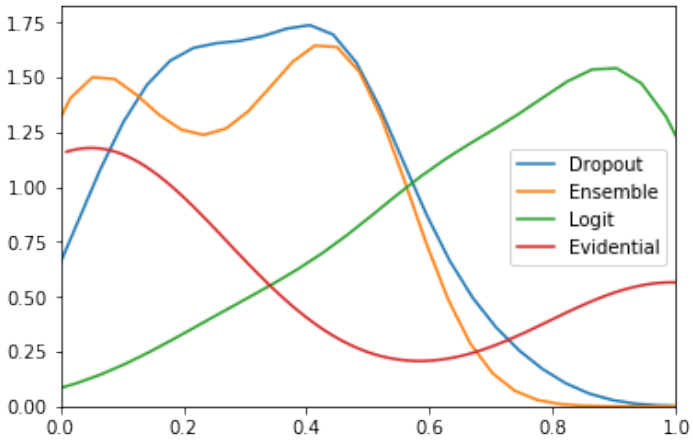


Figure 3. Uncertainty distribution for incorrect predictions on MNIST test set

It is clear the above histograms that other uncertainty methods still tend to make many low uncertainty wrong predictions, whereas our uncertainty method seems to overcome this problem that are often seen in existing methods.

Another test that showed the advantages of our logit uncertainty involves context drift. We used our model that trained on the MNIST data set, and use the model to predict on FashionMNIST data set. FashionMNIST is a data set of fashion images including clothes, dresses, and shoes, with each image of size 28 × 28. We should expect a perfect uncertainty measure to output 1 as uncertainty for all predictions since we are forcing our model to classify the fashion items as digits. Figure 4 shows the distribution of uncertainty values for 60000 training FashionMNIST images using our method and other benchmark methods.

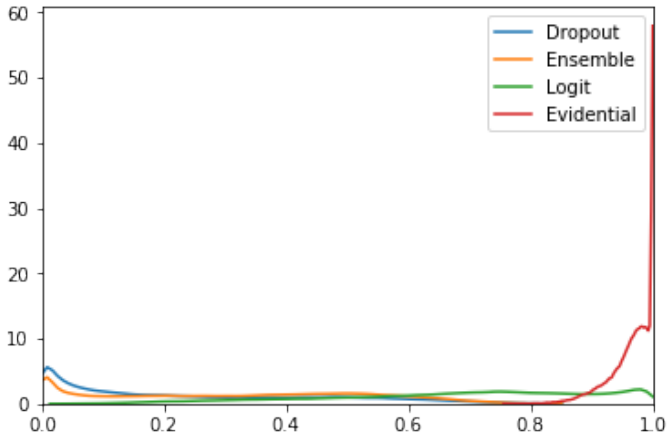


Figure 4. Uncertainty distribution for FashionMNIST

From the results above, although Evidential method seems to have the best performance with the majority of uncertainty values to be above 0.8, we will in the following experiments that Evidential method tends to be overconservative. Compared to Ensemble and Dropout, our Logit uncertainty exhibit the preferred quality that it does not have low uncertainty values, and have many more predictions with high uncertainty values.

The most innovative experiment on our uncertainty measure is to test them on data set that we should expect a perfect uncertainty measure to output intermediate values. To be more specific, for the model trained on MNIST, the handwritten digits data set, we expect the model to output intermediate values for fuzzy handwritten images. To this end, we selected a USPS data set. USPS data set consists of 7291 training images of handwritten digits, however, the crucial difference between the USPS and MNIST is that the size of each USPS image is 16×16 whereas the size of each MNIST image is 28×28. Therefore, if we enlarge the USPS image to 28×28, we expect a perfect uncertainty measure to produce intermediate uncertainty values. Figure 5 shows the uncertainty distribution of our model trained on MNIST and tests them with enlarged USPS images using our logit uncertainty measure and other benchmarks' results.

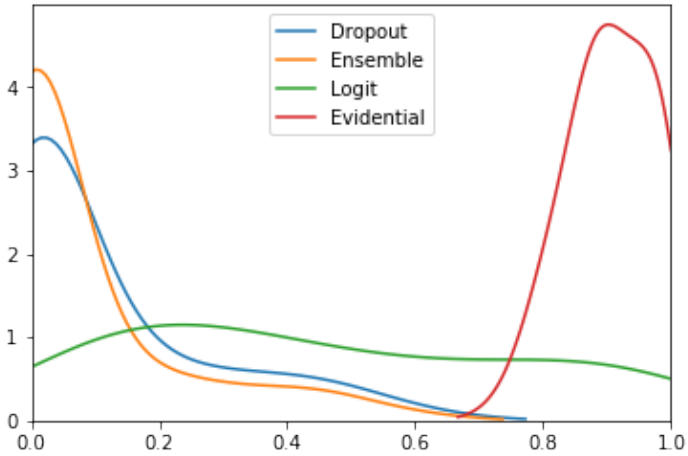


Figure 5. Uncertainty distribution for USPS data set

Compare to the results obtained in Figure 3 and Figure 4 it is obvious that for the USPS test, our uncertainty measure outputs many more uncertainty values from 0.2 to 0.8 range, which

is what we expect for a good uncertainty measure. Other methods have more extreme uncertainty values, and as we explained, Evidential tends to be very conservative and prefer to label predictions with high uncertainty values.

Similar to the previous test, we create another test to see if our uncertainty values are meaningful. We realized that from the beginning to the end of GAN training [7], the generator's images are getting closer and closer to real images from the data set. Therefore, we use a generator with 64 and 128 filters of size 5 × 5 at the first and second convolutional layers to generate images of digits similar to that from the MNIST data set. We trained the GAN for 2000 epochs and let each epoch generate 256 images using the generator. We then use the CNN model trained on MNIST to classify these images and compute their logit uncertainty. Below, we only show the result using GAN to generate images of handwritten digit 7 and results for other digits are similar. Figure 6 shows the average of the 256 uncertainty values after each of the 2000 training epochs.

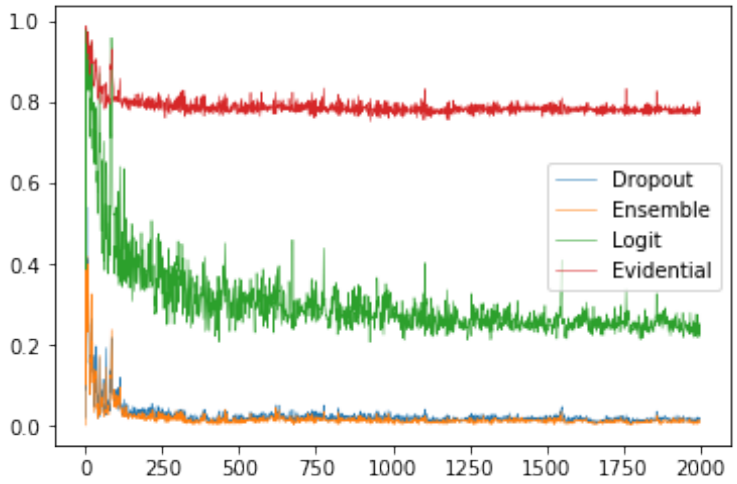
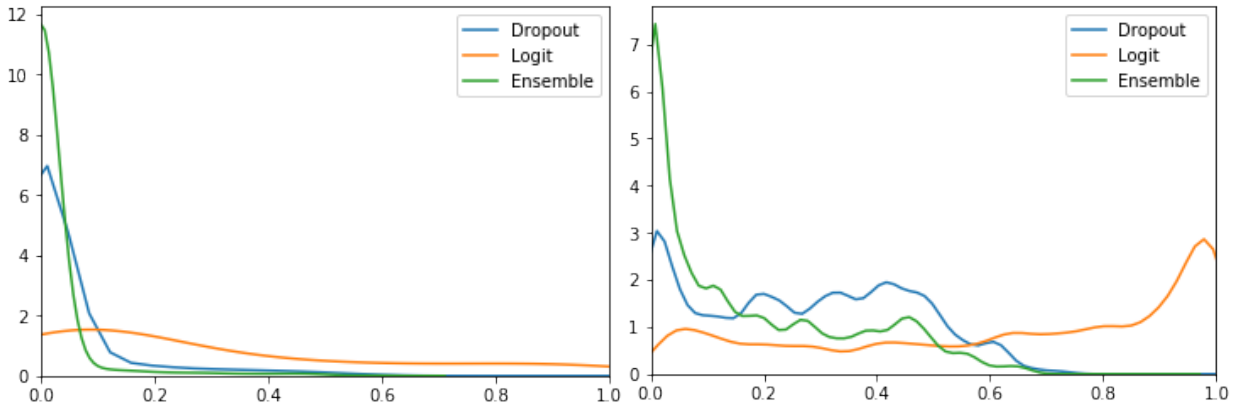


Figure 6. Training epochs vs Average of uncertainty values of images generated after each epoch

We know that during GAN training, generator's images are improving from fuzzy to clear as the training epochs increase, thus we should expect a perfect uncertainty measure to have uncertainty values for these images change from high uncertainty at the start of training to low uncertainty at the end of training. Indeed, that is what we observe in Figure 6 with our logit uncertainty. From the same plot, we observe that the Ensemble and Dropout uncertainty distributions share very different shapes with our logit uncertainty distribution. These two methods' output relatively low uncertainty when the training epoch number is low, which should not be the case for a good uncertainty measure. Evidential, given its conservative tendency, indeed have high uncertainty values even after 2,000 epochs of training.

4.2 Cifar10 data set tests

The Cifar10 data set consists of 60,000 color images of ten object classes and each image is of size 32 × 32. We use a Densenet40 with k = 12 for the training of Cifar10 [9] and then perform similar tests that were used for the MNIST data set. In the first test, we look at the distribution of uncertainty values of correctly and incorrectly classified images of test set (Figure 7).



Uncertainty on correct predictions *Uncertainty on incorrect predictions*

Figure 7. Uncertainty distributions on correctly and incorrectly classified images of Cifar10 test set

The second test involves testing our model trained on Cifar10 on a data set that is very different. For this test, we select the Cifar100 data set, consisting of 60,000 color images of 100 classes, where each image is 32 × 32 in size. The 10 classes of Cifar10 are different from the 100 classes in the Cifar100 data set. Therefore, we should expect a good uncertainty measure to output high uncertainty values, which is what we observe for our method in the uncertainty distribution plot in Figure 8a.

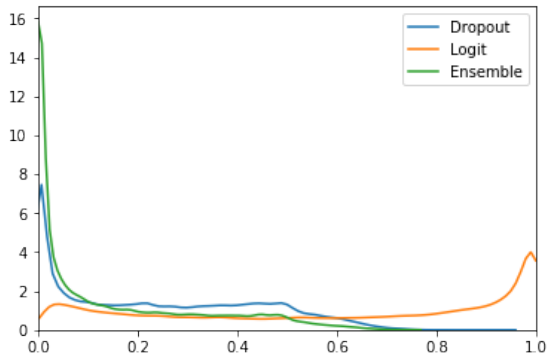


Figure 8a, uncertainty on Cifar100

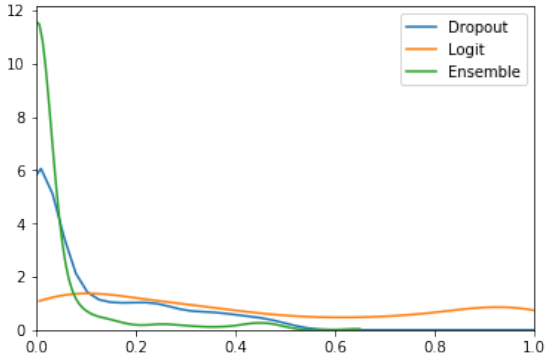


Figure 8b, uncertainty of pickup trucks predicted as auto

The third test for Cifar10 is on images that we should expect a good uncertainty measure to output intermediate values. To achieve this goal, we observed that one of the Cifar10 classes is automobiles with images of passenger cars that do not include pickup trucks. Additionally, Cifar100 has a pickup truck class and, since the pickup trucks and passenger cars are intrinsically similar to each other, we expect the uncertainty values of pickup trucks that are predicted as automobiles to be larger than those of the automobile images in Cifar10 test set yet smaller than those of Cifar100. This is exactly what we have observed as shown in Figure 8b for our Logit uncertainty.

5 Conclusion

This paper proposed a new uncertainty measure for neural network predictions. Our measure is general in the sense that it is agnostic to the network architecture, to the learning procedure, and to the training task. Our extensive experiments showed our method's superior performance in different tasks with different architectures compare to existing uncertainty methods.

The innovative GAN tests demonstrated the intermediate uncertainty values and their order relationships are meaningful, which shows a significant improvement from existing methods. For future research, we hope to continue testing our uncertainty measure in more settings and evaluate the performance of our uncertainty measure in real world applications.

References:

[1]. Hugo Touvron, Andrea Vedaldi, Matthijs Douze, and Herve Jegou. Fixing the train-test resolution discrepancy. *arXiv preprint arXiv 1906.06423*, 2019

[2]. Heinrich Jiang, Been Kim, Melody Y. Guan, and Maya Gupta. To Trust or Not to Trust A Classifier. *NIPS*, 2018

[3]. Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *NIPS,* 2017.

[4]. Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. *In International Conference on Machine Learning, pages 1050–1059*, 2016.

[5]. Murat Sensoy, Lance Kaplan, and Melih Kandemir. Evidential Deep Learning to Quantify Classification Uncertainty. *NIPS*, 2018

[6]. Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight Uncertainty in Neural Networks. *In International Conference on Machine Learning,* 2015

[7]. Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. *NIPS*, 2014

[8]. Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. Training Confidence-calibrated classifiers for detecting out-of-distribution samples. *ICLR*, 2018

[9]. Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely Connected Convolutional Networks. *arXiv preprint arXiv 1608.06993v5*. 2018

[10]. Rob J. Hyndman. Computing and Graphing High Density Regions. *The American Statistician. Vol. 50, No. 2. pp. 120-126*. 1996