Principled Uncertainty Estimates for Adversarial Robustness

Subhankar Chakraborty
Sumanth R Hegde
Sourav Sahoo

Motivation

- Uncertainty information is crucial in areas such as medical sciences, autonomous driving and to safeguard a model from an adversary.
- Bayesian methods are traditionally used - hard to optimise!
- MC Dropout : uncertainty estimates from standard neural nets with dropout
- Recent works based on Evidence theory have been shown to be more reliable than traditional Bayesian methods.

The story so far ...

- We proved that a neural network with dropout is an approximation of a Gaussian Process with two components with other appropriate conditions
- We showed that such a network is uncertain in its prediction for a data point far away from the training dataset [Solar irradiance dataset, CO_2 dataset] as expected
- We proposed to tackle adversarial examples based on model uncertainty using Monte-Carlo dropout and other methods.

But, there is a problem ...

- The distribution of the layer weights is a two component Gaussian mixture as shown below:

$$q(\mathbf{w_q}) = p\mathcal{N}(\mathbf{m}_q, \sigma^2 I_K) + (1-p)\mathcal{N}(0, \sigma^2 I_K)$$

- The variances of the individual normal components are set to be almost zero under the GP assumption.
- So, the model variance (uncertainty) is solely due to the variance of a Bernoulli random variable with parameter ***p*** i.e. the dropout parameter.
- BUT, we are yet to find the optimal value of ***p*** that determines the uncertainty estimate of the model !

Concrete Dropout

Automatic tuning of dropout probabilities

Nomenclature

- **Epistemic Uncertainty**: It refers to the uncertainty of model that can be explained away with enough number of data points.

- **Aleatoric Uncertainty**: It captures noise inherent in the observations. This could be for example sensor noise or motion noise, resulting in uncertainty which **cannot** be reduced even if more data were to be collected.

- **Predictive Uncertainty**: It is obtained by combining the above two uncertainties. It presents the model's confidence in its prediction, taking into account noise it can explain away and noise it cannot.

Concrete Dropout : The Motivation

- A naive approach is to conduct grid search over the dropout parameters. But grid-searching over the dropout probability can be expensive and time consuming, especially when done with large models.
- Instead, the dropout probability can be optimised using a gradient method, where we seek to minimise some objective with respect to (w.r.t.) that parameter.
- This concept was introduced by Gal et al. in their paper “Concrete Dropout”.

The Optimization Objective : The Motivation

- Earlier, to approximate a neural network with dropout as a GP, we had shown:

We would like our approximating distribution to be as close as possible to the posterior distribution obtained from the full Gaussian process. We thus minimise the Kullback–Leibler (KL) divergence, intuitively a measure of similarity between two distributions:

$$\mathrm{KL}(q(\omega) \mid p(\omega|\mathbf{X}, \mathbf{Y})),$$

resulting in the approximate predictive distribution

$$q(\mathbf{y}^*|\mathbf{x}^*) = \int p(\mathbf{y}^*|\mathbf{x}^*, \omega) q(\omega) \mathrm{d}\omega. \tag{6}$$

Minimising the Kullback–Leibler divergence is equivalent to maximising the *log evidence lower bound* [Bishop, 2006],

$$\mathcal{L}_{\mathrm{VI}} := \int q(\omega) \log p(\mathbf{Y}|\mathbf{X}, \omega) \mathrm{d}\omega - \mathrm{KL}(q(\omega)||p(\omega)) \tag{7}$$

The Optimization Objective

- Following dropout's variational interpretation, the optimization objective is chosen as:

$$\widehat{\mathcal{L}}_{\mathrm{MC}}(\theta) = -\frac{1}{M}\sum_{i\in S}\log p(\mathbf{y}_i|\mathbf{f}^{\boldsymbol{\omega}}(\mathbf{x}_i)) + \frac{1}{N}\mathrm{KL}(q_\theta(\boldsymbol{\omega})||p(\boldsymbol{\omega}))$$

- The second term can be simplified to:

$$\mathrm{KL}(q_\theta(\boldsymbol{\omega})||p(\boldsymbol{\omega})) = \sum_{l=1}^{L}\mathrm{KL}(q_{\mathbf{M}_l}(\mathbf{W}_l)||p(\mathbf{W}_l))$$

$$\mathrm{KL}(q_{\mathbf{M}}(\mathbf{W})||p(\mathbf{W})) \propto \frac{l^2(1-p)}{2}||\mathbf{M}||^2 - K\mathcal{H}(p)$$

Regular Entropy Term which is also the *dropout regularizer*.

Weight regularizer

The Concrete Relaxation

- Numerous gradient estimators are present to compute the derivative but pathwise derivative estimator (PDE)(a.k.a *re-parametrization trick* or *stochastic backpropagation*) is used as it has a low variance in estimation.
- But, PDE assumes the distribution can be re-parameterized as $g(\boldsymbol{\theta}, \in)$ where $\boldsymbol{\theta}$ is the distribution parameters and $\in$ is a random variable independent of $\boldsymbol{\theta}$. But, the Bernoulli distribution doesn't admit such re-parametrization.
- So, Concrete distribution relaxation is used which is an continuous relaxation of discrete Bernoulli random variable. For a Bernoulli r.v., the Concrete relaxation has a closed-form expression as follows...

The Concrete Relaxation

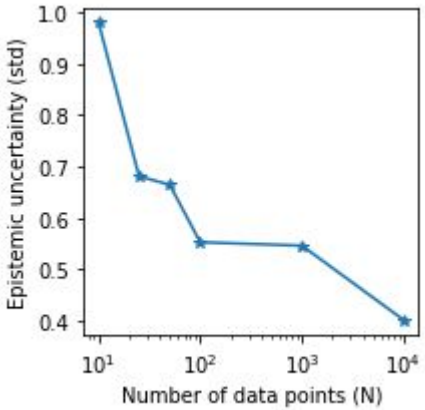
$$\tilde{\mathbf{z}} = \text{sigmoid}\left(\frac{1}{t} \cdot \left(\log p - \log(1-p) + \log u - \log(1-u)\right)\right)$$

- u ~ Uniform[0,1], t is called temperature
- The random variable is in interval [0,1](the range of sigmoid function) and most mass is concentrated around the boundaries.
- So, it can be considered as a good proxy for the Bernoulli r.v.

Experimental Setup

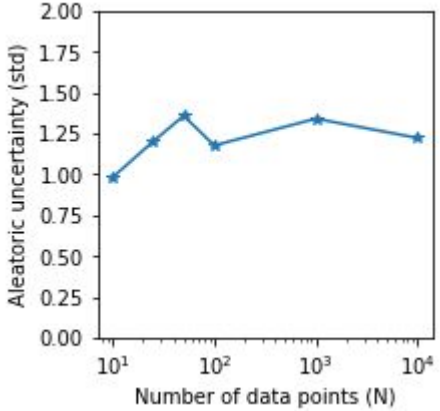
- We built a synthetic dataset $y = 2\sin(x) + 8 + \epsilon$, where $\epsilon \sim \mathcal{N}(0,1)$ with 10k data points.
- The neural network architecture we chose to implement Concrete dropout is a three layer fully connected network, each having 512 units.
- We trained the model starting from 10 data points till all the way upto 10k data points.
- The epistemic, aleatoric and predictive uncertainty is recorded.

Experimental Results



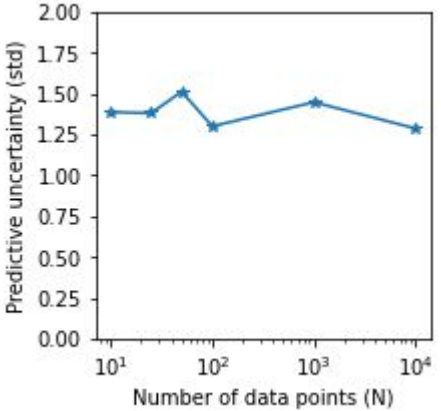
(a) Epistemic Uncertainty

(got by multiple forward passes with dropout, as in training time)



(b) Aleatoric Uncertainty

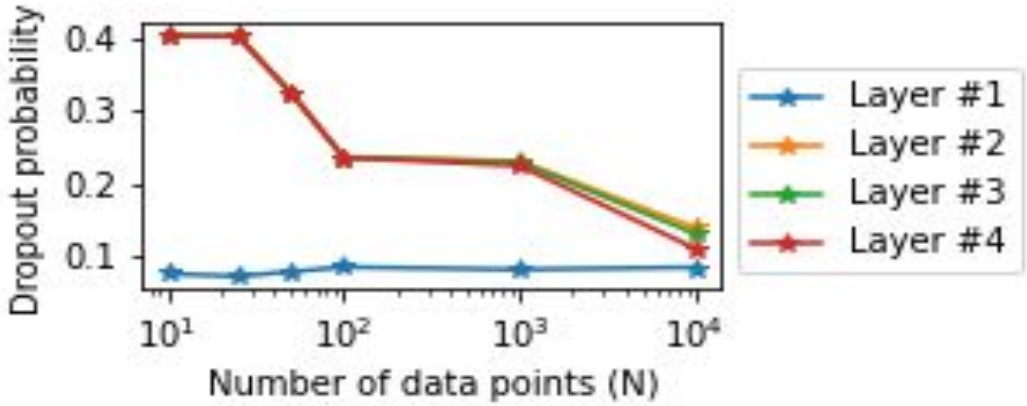
(got by heteroscedastic regression)



(c) Predictive Uncertainty

(combination of previous two)

Experimental Results



The final dropout probabilities of models trained with different number of data points. It is to be noted that the dropout probabilities for the top layers decrease as the number of data points increase. This shows that the model only becomes confident about the optimal input transformation (dropout probabilities are set to zero) after seeing a relatively large number of examples.

Using uncertainty estimates for detecting adversarial examples

Which uncertainty measures make sense? Which uncertainties are reliable?

The quest for adversarial robustness

- Szegedy et. al showed that state-of-the-art neural networks are susceptible to adversarial examples i.e. images with a carefully calculated perturbation added, which can cause the model to give erroneous, over-confident results.

- This phenomenon has been shown to exhibit *cross model generalization* and *cross training set generalization.*

- Naive adversarial training, in which a network is trained on adversarial examples, is not an effective way for adversarial robustness.

- Bayesian methods provide a natural way of detecting adversaries through predictive uncertainty.

Predictive Entropy

- If the model output is $P(y|x)$ over the set of outcomes $\mathcal{Y}$, the predictive entropy is

$$H[P(y|x)] = -\sum_{y \in \mathcal{Y}} P(y|x) \log P(y|x).$$

- However, this does not distinguish between epistemic and aleatoric uncertainty.

- Doing so may be useful : We want to capture when the input lies far away from the data space, where the model would be poorly constrained; and distinguish this from a scenario where the input lies near the data space but with noisy labels.

Mutual Information

- The Mutual Information (MI) between random variables **X** and **Y** is defined as

$$I(X,Y) = H[P(X)] - \mathbb{E}_{P(y)} H[P(X \mid Y)]$$
$$= H[P(Y)] - \mathbb{E}_{P(x)} H[P(Y \mid X)].$$

- The amount of information we would gain about the model parameters if we were to receive a label $\mathscr{Y}$ for a new point x, given the dataset $\mathcal{D}$ is then given by

$$I(\omega, y \mid \mathcal{D}, x) = H[p(y \mid x, \mathcal{D})] - \mathbb{E}_{p(\omega \mid \mathcal{D})} H[p(y \mid x, \omega)]$$

- Being uncertain about an input point x implies that if we knew the label at that point we would gain information.

Mutual Information

- Conversely, if the parameters at a point are already well determined, then we would gain little information from obtaining the label.
- Thus, the MI is a measurement of the model's epistemic uncertainty.
- In the form presented above, it is also readily approximated using the Bayesian interpretation of dropout.
- The first term we will refer to as the **predictive entropy**; this is just the entropy of the predictive distribution.
- The second term is the mean of the entropy of the predictions given the parameters over the posterior distribution $p(\omega \mid \mathcal{D})$, and we thus refer to it as the **expected entropy**.

Mutual Information (MI)

- These quantities are not tractable analytically for deep nets, but using the dropout approximation and the Monte Carlo estimator , the different terms are readily computable:

$$
\begin{aligned}
p(y \mid \mathcal{D}, \mathbf{x}) &\simeq \frac{1}{T} \sum_{i=1}^{T} p(y \mid \omega_i, \mathbf{x}) \\
&:= p_{MC}(y \mid \mathbf{x}) \\
H[p(y \mid \mathcal{D}, \mathbf{x})] &\simeq H[p_{MC}(y \mid \mathcal{D}, \mathbf{x})] \\
I(\omega, y \mid \mathcal{D}, x) &\simeq H[p_{MC}(y \mid \mathcal{D}, \mathbf{x})] \\
&\quad - \frac{1}{T} \sum_{i=1}^{T} H[p(y \mid \omega_i, \mathbf{x})]
\end{aligned}
$$

where $\omega_i \sim q(\omega \mid \mathcal{D})$ are samples from the dropout distribution.

A Note on Softmax Variance

- Another ad-hoc measure of uncertainty is the softmax variance, ie, the variance of the softmax probabilities $p(y = c \mid \omega_i, \mathbf{x})$ with the variance calculated over i) which can be seen as a proxy to the mutual information.
- We can observe that the variance is the leading term in the series expansion of the mutual information.
- The variance score is the mean variance across the classes

$$\hat{\sigma}^2 = \frac{1}{C}\sum_{j=1}^{C}\frac{1}{T}\sum_{i=1}^{T}(p_{ij} - \hat{p}_j)^2$$

$$= \frac{1}{C}\left(\sum_{j=1}^{C}\left(\frac{1}{T}\sum_{i=1}^{T}p_{ij}^2\right) - \hat{p}_j^2\right)$$

A Note on Softmax Variance

- The Mutual Information Score is

$$\hat{I} = H(\hat{p}) - \frac{1}{T}\sum_i H(p_i)$$

$$= \sum_j \left(\frac{1}{T}\sum_i p_{ij} \log p_{ij} \right) - \hat{p}_j \log \hat{p}_j$$

- Using a Taylor expansion of the logarithm

$$\hat{I} = \sum_j \left(\frac{1}{T}\sum_i p_{ij}(p_{ij} - 1) \right) - \hat{p}_j(\hat{p}_j - 1) + \ldots$$

$$= \sum_j \left(\frac{1}{T}\sum_i p_{ij}^2 \right) - \hat{p}_j^2 - \left(\frac{1}{T}\sum_i p_{ij} \right) + \hat{p}_j + \ldots$$

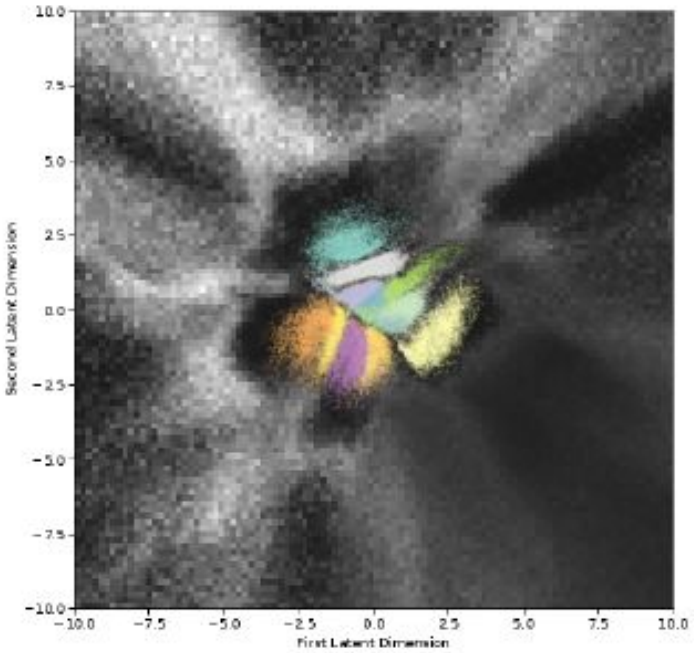
$$= \sum_j^C \left(\frac{1}{T}\sum_i^T p_{ij}^2 \right) - \hat{p}_j^2 + .$$

- We see that the first term in the series is identical up to a multiplicative constant to the mean variance of the samples.

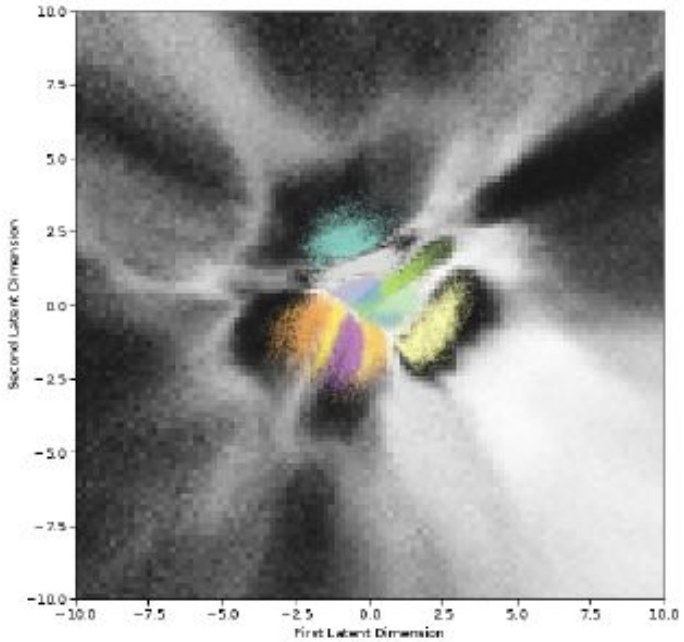
Evaluating uncertainty measures

- Experiment : The MNIST dataset is compressed onto a 2-dimensional latent space using a variational auto-encoder.
- For every point in the latent space, the corresponding image is decoded to obtain model predictions.
- Different uncertainty measures (predictive entropy, MI) are evaluated and visualised.
- For points far away from those corresponding to training data, the uncertainty should be high.
- These should be distinguishable from ambiguous images i.e. images which resemble digits but are inherently ambiguous.

Note : Different colours indicate training data belonging to different classes of the MNIST dataset. A lighter background implies higher uncertainty.



Mutual information



Predictive entropy

Uncertainty measures evaluated for a standard dropout network

Why MC dropout fails

- Question : How well does the dropout approximation capture uncertainty ?
- Variational inference schemes typically underestimate the *uncertainty of the posterior*, tending to fit an approximation to a local mode rather than capturing the full posterior.
- An MC dropout network's uncertainty has 'holes'.
- These are regions where such models are mistakenly overconfident, which adversarial attack algorithms can exploit.
- We can capture the posterior using an ensemble of dropout models using different initializations, assuming that these will converge to different local modes.

Evidential Deep Learning to Quantify Model Uncertainty

A *Theory of Evidence* Perspective

In a Nutshell.....

- Traditional Bayesian methods model probability distributions over the weights/ parameters of the neural network.
- Here, predictive uncertainties are inferred directly by placing a Dirichlet distribution over the class probabilities.
- The model's outputs are treated as *subjective opinions* and we learn the function that collects the *evidence* that back these opinions.
- Effectively tackles out-of-distribution queries : the model is able to simply say "I do not know" for test samples belonging to an unrelated data distribution.

The problems with softmax

- Softmax outputs can be interpreted as the parameters of a categorical distribution

$$Pr(y|\mathbf{x},\theta) = \text{Mult}(y|\sigma(f_1(\mathbf{x},\theta)),\cdots,\sigma(f_K(\mathbf{x},\theta))),$$

- Due to the exponent used, the probabilities are often blown up, making them unreliable for reliable uncertainty estimates.
- In this work, the model outputs are treated as parameters of a Dirichlet distribution, which is a “meta” probability distribution i.e. a distribution over possible class probabilities.

The Dempster-Shafer Theory of Evidence

- We start with a *frame of discernment* : the set of *possibilities* in consideration.
- Assign belief masses (subjective probabilities) to any subset of the frame, including the frame itself.
- In case all the masses are assigned to the whole frame, then that represents a state of complete uncertainty (“I do not know”)
- For a set $X = \{x_1, ..., x_K\}$, we assign a belief masses $\mathbf{b^X}$ to each element and an uncertainty u^X which represents vacuity of evidence.
- These K+1 quantities sum to one i.e.

$$u + \sum_{k=1}^{K} b_k = 1,$$

The Dempster-Shafer Theory of Evidence

- Each probability estimate is thus treated as a *subjective opinion*, with u^X explicitly modelling the certainty/uncertainty in our estimate.
- Each belief mass b_k assigned is based on evidence e_k collected for the kth class.

$$b_k = \frac{e_k}{S} \quad \text{and} \quad u = \frac{K}{S},$$

Where $S = \sum_{i=1}^{K} (e_i + 1).$

- Each belief mass assignment corresponds to a Dirichlet distribution with parameters $\alpha_k = e_k + 1.$

The Dirichlet Distribution

- The Dirichlet distribution with parameters $\boldsymbol{\alpha} = \{\alpha_1,.....,\alpha_K\}$ is given by

$$D(\mathbf{p}|\boldsymbol{\alpha}) = \begin{cases} \frac{1}{B(\boldsymbol{\alpha})} \prod_{i=1}^{K} p_i^{\alpha_i - 1} & \text{for } \mathbf{p} \in \mathcal{S}_K, \\ 0 & \text{otherwise,} \end{cases}$$

Where S_K is the K-dimensional unit simplex,

$$\mathcal{S}_K = \left\{ \mathbf{p} \middle| \sum_{i=1}^{K} p_i = 1 \text{ and } 0 \leq p_1, \ldots, p_K \leq 1 \right\}$$

and $B(\boldsymbol{\alpha})$ is the multinomial beta function.

The Dirichlet Distribution

- Assume for a 10-class classification problem, we have **b =** (0,0,.....0) as the belief mass assignment.
- We have no evidence in support of the data sample belonging to ANY particular class.
- Thus, the corresponding Dirichlet distribution has parameters = (1,1,...1) => Uniform distribution (over class probabilities)
- This is a case of total uncertainty, which is captured by the uncertainty u being 1.

Learning to form opinions

- For a given set of parameters $\boldsymbol{\alpha}$, the expected class probabilities are :

$$\hat{p}_k = \frac{\alpha_k}{S}.$$

- For each training sample i, we wish to learn multinomial opinions for their classification, through a Dirichlet distribution.

Let $\mathbf{y}_i$ be a one-hot vector encoding the ground-truth class of observation $\mathbf{x}_i$ with $y_{ij} = 1$ and $y_{ik} = 0$ for all $k \neq j$, and $\boldsymbol{\alpha}_i$ be the parameters of the Dirichlet density on the predictors. First, we can treat $D(\mathbf{p}_i|\boldsymbol{\alpha}_i)$ as a prior on the likelihood $\text{Mult}(\mathbf{y}_i|\mathbf{p}_i)$ and obtain the negated logarithm of the marginal likelihood by integrating out the class probabilities

$$\mathcal{L}_i(\Theta) = -\log\left(\int \prod_{j=1}^{K} p_{ij}^{y_{ij}} \frac{1}{B(\boldsymbol{\alpha}_i)} \prod_{j=1}^{K} p_{ij}^{\alpha_{ij}-1} d\mathbf{p}_i\right) = \sum_{j=1}^{K} y_{ij}\Big(\log(S_i) - \log(\alpha_{ij})\Big) \qquad (3)$$

and minimize with respect to the $\boldsymbol{\alpha}_i$ parameters. This technique is well-known as the Type II Maximum Likelihood.

Learning to form opinions

The sum of squares loss was found to perform better empirically :

$$\begin{aligned}\mathcal{L}_i(\Theta) &= \int ||\mathbf{y}_i - \mathbf{p}_i||_2^2 \frac{1}{B(\boldsymbol{\alpha}_i)} \prod_{i=1}^{K} p_{ij}^{\alpha_{ij}-1} d\mathbf{p}_i \\ &= \sum_{j=1}^{K} \mathbb{E}\left[y_{ij}^2 - 2y_{ij}p_{ij} + p_{ij}^2\right] = \sum_{j=1}^{K} \left(y_{ij}^2 - 2y_{ij}\mathbb{E}[p_{ij}] + \mathbb{E}[p_{ij}^2]\right). \end{aligned} \tag{5}$$

Learning to form opinions

The first advantage of the loss in Equation 5 is that using the identity

$$\mathbb{E}[p_{ij}^2] = \mathbb{E}[p_{ij}]^2 + \mathrm{Var}(p_{ij}),$$

we get the following easily interpretable form

$$\begin{aligned}
\mathcal{L}_i(\Theta) &= \sum_{j=1}^{K} (y_{ij} - \mathbb{E}[p_{ij}])^2 + \mathrm{Var}(p_{ij}) \\
&= \sum_{j=1}^{K} \underbrace{(y_{ij} - \alpha_{ij}/S_i)^2}_{\mathcal{L}_{ij}^{err}} + \underbrace{\frac{\alpha_{ij}(S_i - \alpha_{ij})}{S_i^2(S_i+1)}}_{\mathcal{L}_{ij}^{var}} \\
&= \sum_{j=1}^{K} (y_{ij} - \hat{p}_{ij})^2 + \frac{\hat{p}_{ij}(1-\hat{p}_{ij})}{(S_i+1)}.
\end{aligned}$$

Learning to form opinions

- In order to ensure that the model learns to output the uniform distribution in cases of low evidence, a regularisation term is added. The total loss is :

$$\mathcal{L}(\Theta) = \sum_{i=1}^{N} \mathcal{L}_i(\Theta) + \lambda_t \sum_{i=1}^{N} KL[D(\mathbf{p_i}|\tilde{\boldsymbol{\alpha}}_i) \,||\, D(\mathbf{p}_i|\langle 1, \ldots, 1\rangle)],$$

- Reasoning : During training, the model may discover patterns in the data and generate evidence for specific class labels.
- For instance, the model may discover that the existence of a large circular pattern on MNIST images may lead to evidence for the digit zero.
- This means that the output i.e., the evidence for class label 0, should be increased when such a pattern is observed.

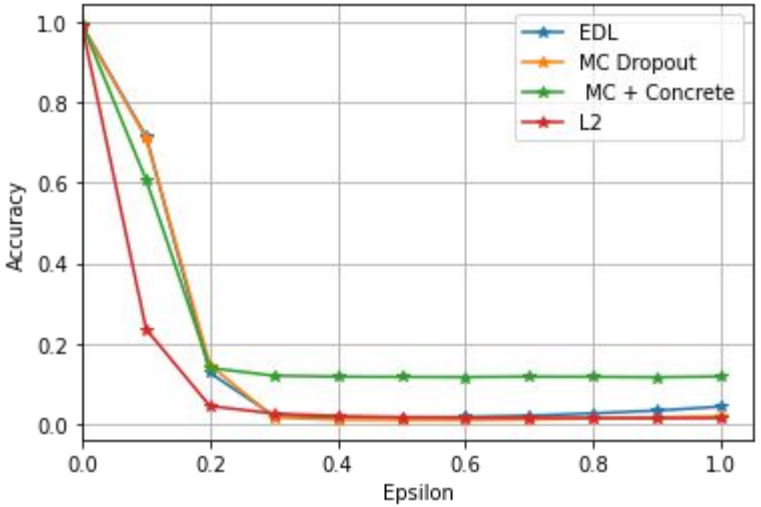
Learning to form opinions

- However, when counter-examples are observed (e.g., a digit six with the same circular pattern), the parameters of the neural network should be tuned by back propagation to generate smaller amounts of evidence for this pattern and minimize the loss of these samples, as long as the overall loss also decreases.
- This may not happen with lesser counter examples, leading to evidence generated for incorrect assignments

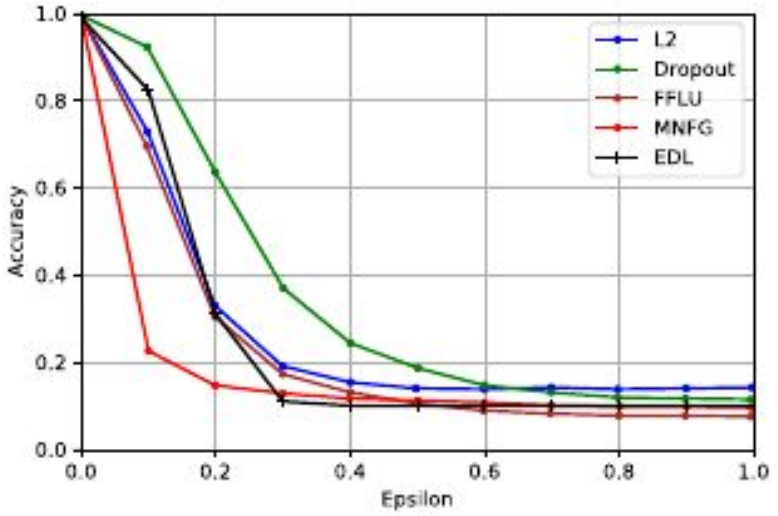
Evaluation on Adversarial examples

- Here, we make a comparative evaluation of different models on adversarial examples generated from the MNIST test set using the Fast Gradient Sign Method.
- The common architecture used is the LeNet variant in the Evidential Deep Learning (EDL) paper. Four different models are compared :
  - L2 : LeNet network trained with softmax output and L2 regularisation.
  - MC Dropout : The Monte-Carlo Dropout method.
  - MC + Concrete : Same as MC Dropout, but with Concrete Dropout layer used.
  - EDL : The model in *Evidential Deep Learning* paper discussed previously.

Experimental Results

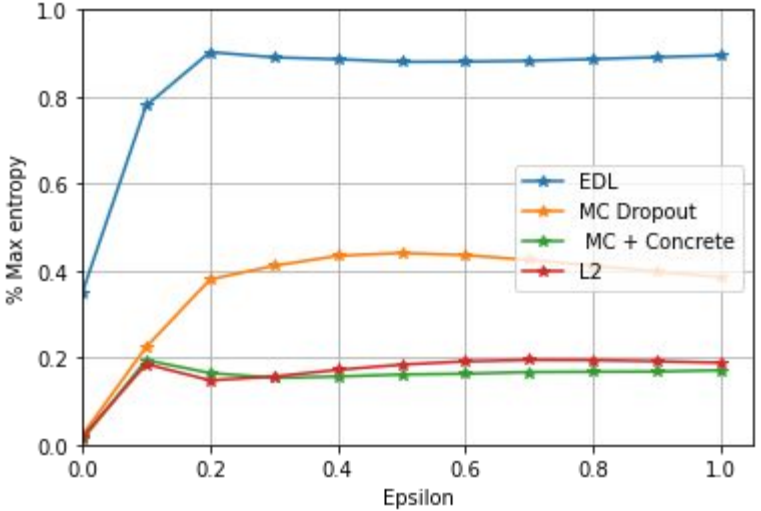


Accuracy vs perturbation ϵ on MNIST.
Experimental

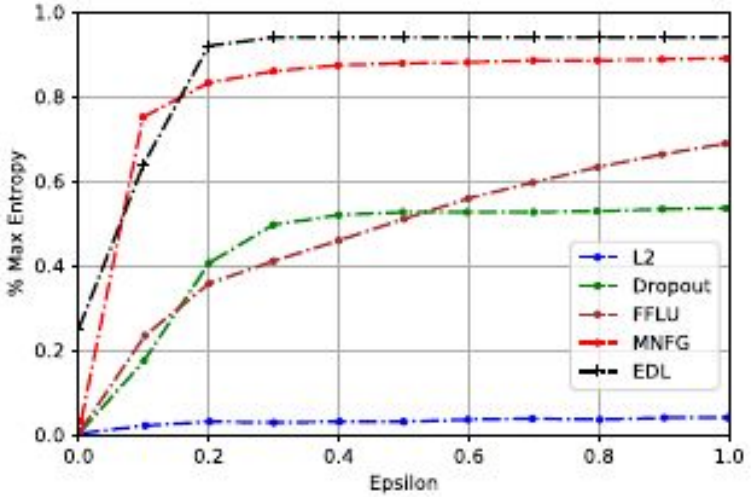


Accuracy vs perturbation ϵ on MNIST.
Results quoted in EDL

Experimental Results

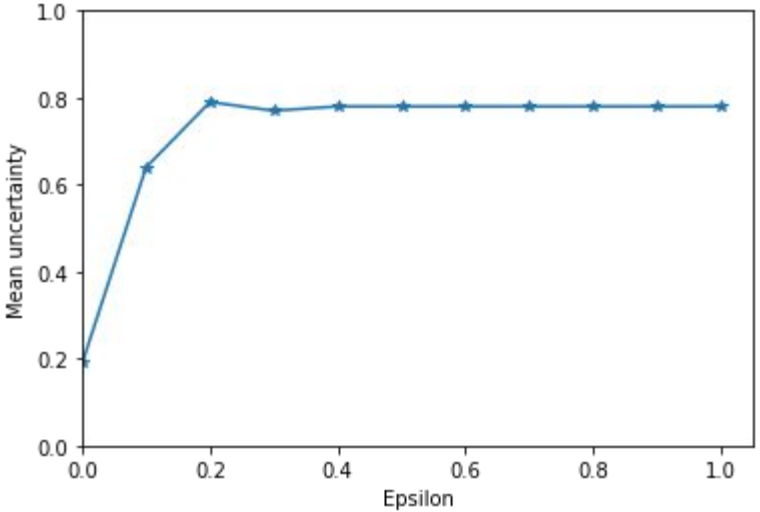


Entropy vs perturbation ϵ on MNIST.
Experimental



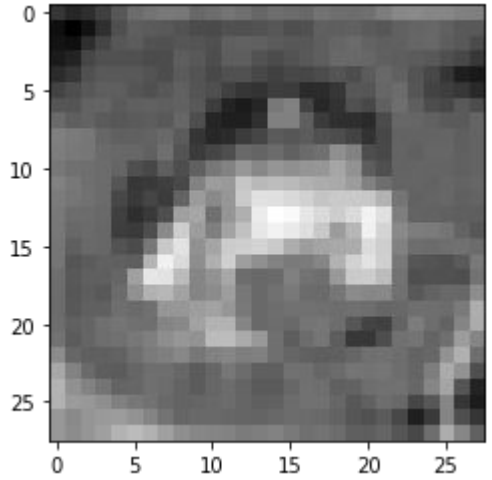
Entropy vs perturbation ϵ on MNIST.
Results quoted in EDL

Experimental Results



Mean uncertainty vs perturbation ϵ on MNIST test set.
Experimental

Can the model say "*I do not know*" ?



An image of a frog from the CIFAR10 dataset (grayscale)

Softmax probabilities by L2 :

[0. 0. 0. 0. 0. 0. 0. 0. 1. 0.] => Classified as 8!

Output by EDL :

Class probabilities : [0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1]

Uncertainty : 1.0

Conclusions

- There is the fundamental question of whether adversarial examples are an intrinsic property of neural nets or they are an artefact which can be addressed through better training procedures.
- Concrete dropout provides better uncertainty estimation but does not perform well in adversarial attacks.
- MC dropout nets have the tendency to capture the local behaviour of the distribution around a mode rather than the full posterior.
- They are vulnerable to adversarial attacks since they do not capture the full posterior and there are regions where they are mistakenly overconfident.
- Evidence theory looks to be a promising direction for robust neural networks.

References

- Gal, Yarin et al. “Concrete Dropout.” *NIPS* (2017).
- Szegedy, Christian, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow and Rob Fergus. “Intriguing properties of neural networks.” *CoRR* abs/1312.6199 (2014): n. pag.
- Smith, Lewis, and Yarin Gal. "Understanding measures of uncertainty for adversarial example detection." *arXiv preprint arXiv:1803.08533* (2018).
- Sensoy, Murat, Lance Kaplan, and Melih Kandemir. "Evidential deep learning to quantify classification uncertainty." In *Advances in Neural Information Processing Systems*, pp. 3179-3189. 2018.

Questions?