VIDY Reading Group
Uncertainty Quantification: Theory & Algorithms

Tuo Wang
Computer Science
Virginia Tech

Dec 27, 2023

Roadmap

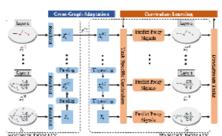
Background

Motivation Overview

I. Background and Motivations

- What is uncertainty?
- Why is is important
- Existing approaches

Paper #1



II. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning

Paper #2

III. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles

Conclusion

IV. Conclusion

What is uncertainty?

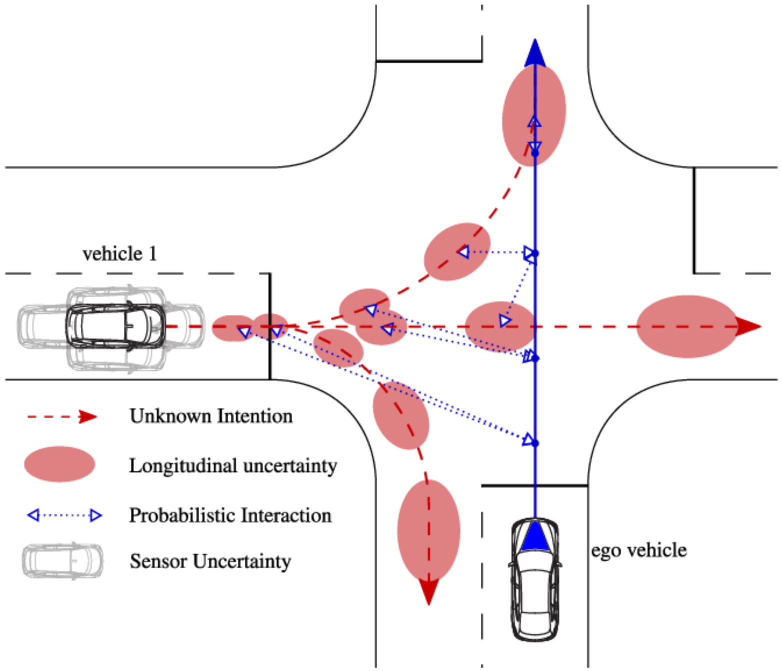
- First brought by **Frank Knight** in 1921
- **Uncertainty** is the inability to forecast the likelihood of events happening in the future.

Frank Hyneman Knight (November 7, 1885 – April 15, 1972) was an American economist who spent most of his career at the University of Chicago. He is best known as the author of the book Risk, Uncertainty and Profit (1921), based on his PhD dissertation at Cornell University.

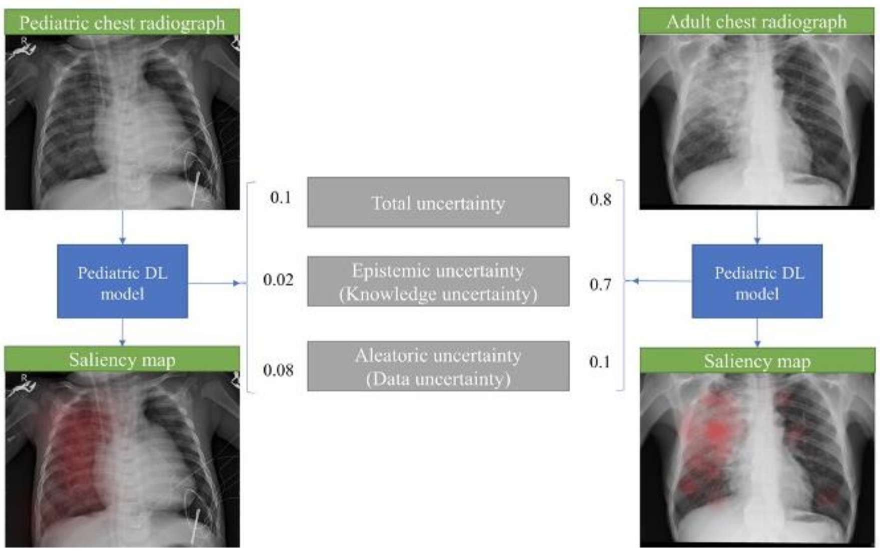
Knight, Frank Hyneman. *Risk, uncertainty and profit*. Vol. 31. Houghton Mifflin, 1921.

Why is it important?

- Uncertainty measures the reliability of model predictions.
- Many applications require safe and reliable predictions, and hence a certain level of self-awareness.



A typical situation in which the ego vehicle has to decide for an action[1]

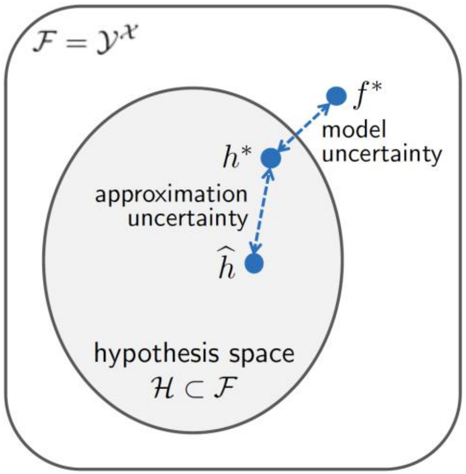


A typical situation in which the diagnosis model has to give prediction[2]

1. Hubmann, Constantin et al. "Automated Driving in Uncertain Environments: Planning With Interaction and Uncertain Maneuver Prediction." *IEEE Transactions on Intelligent Vehicles* 3 (2018): 5-17.
2. Faghani S, Moassefi M, Rouzrokh P, Khosravi B, Baffour FI, Ringler MD, Erickson BJ. Quantifying Uncertainty in Deep Learning of Radiologic Images. Radiology. 2023 Aug;308(2):e222217. doi: 10.1148/radiol.222217. PMID: 37526541.

Where does uncertainty come from?

- Epistemic, reducible
  - Hypothesis
  - Approximation
- Aleatoric, irreducible
  - Data



$$f^*(\boldsymbol{x}) := \arg\min_{\hat{y} \in \mathcal{Y}} \int_{\mathcal{Y}} \ell(y, \hat{y})\, d\mathbf{P}(y \mid \boldsymbol{x})$$

h* is the hypothesis chosen from the hypothesis space

ĥ is the hypothesis of a learner, which aims to estimate h*

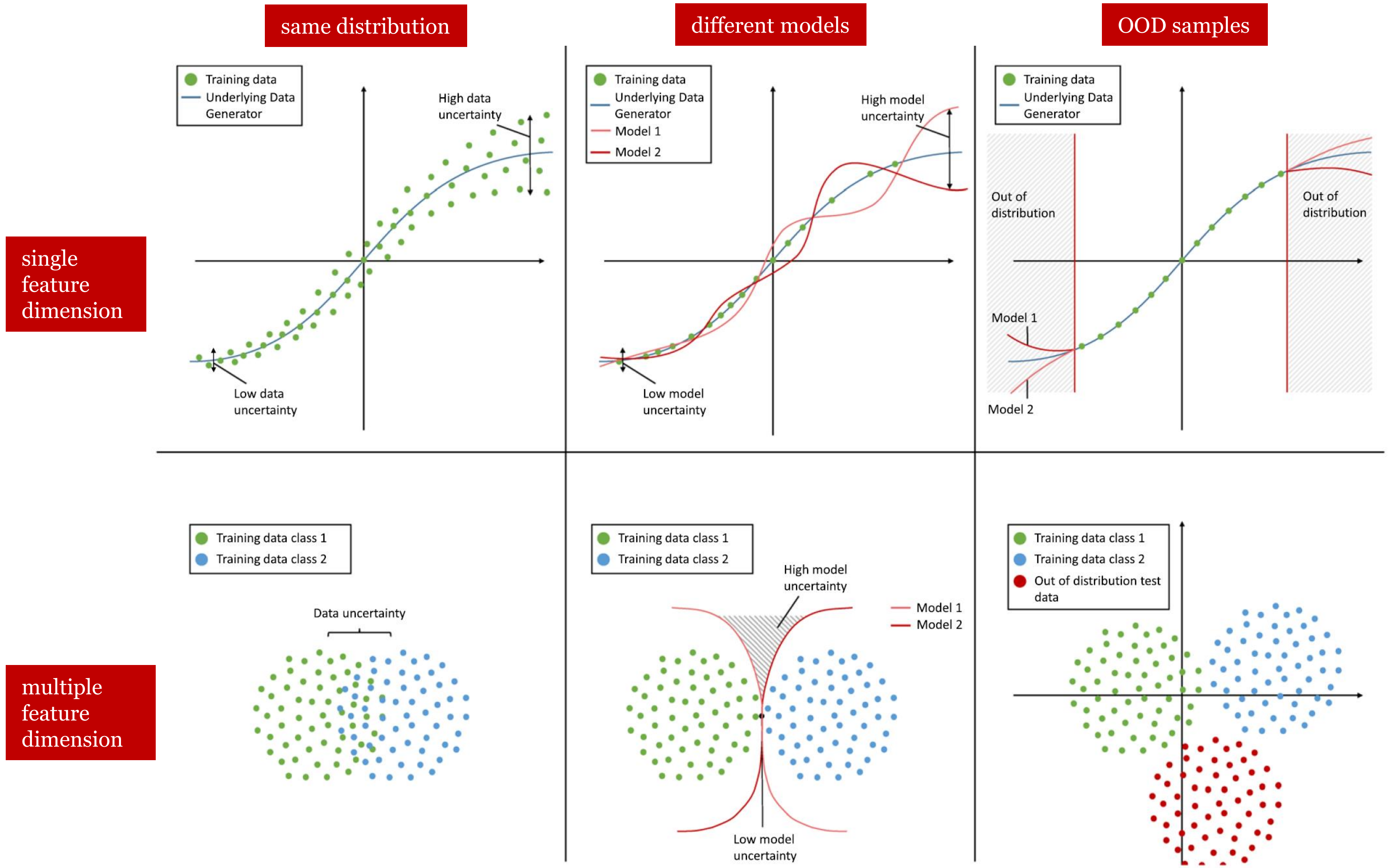
	point prediction	probability
ground truth	$f^*(\boldsymbol{x})$	$\mathbf{p}(\cdot \mid \boldsymbol{x})$
best possible	$h^*(\boldsymbol{x})$	$\mathbf{p}(\cdot \mid \boldsymbol{x}, h^*)$
induced predictor	$\hat{h}(\boldsymbol{x})$	$\mathbf{p}(\cdot \mid \boldsymbol{x}, \hat{h})$

PU = **EU** + **AU**

predictive uncertainty = epistemic uncertainty + aleatoric uncertainty

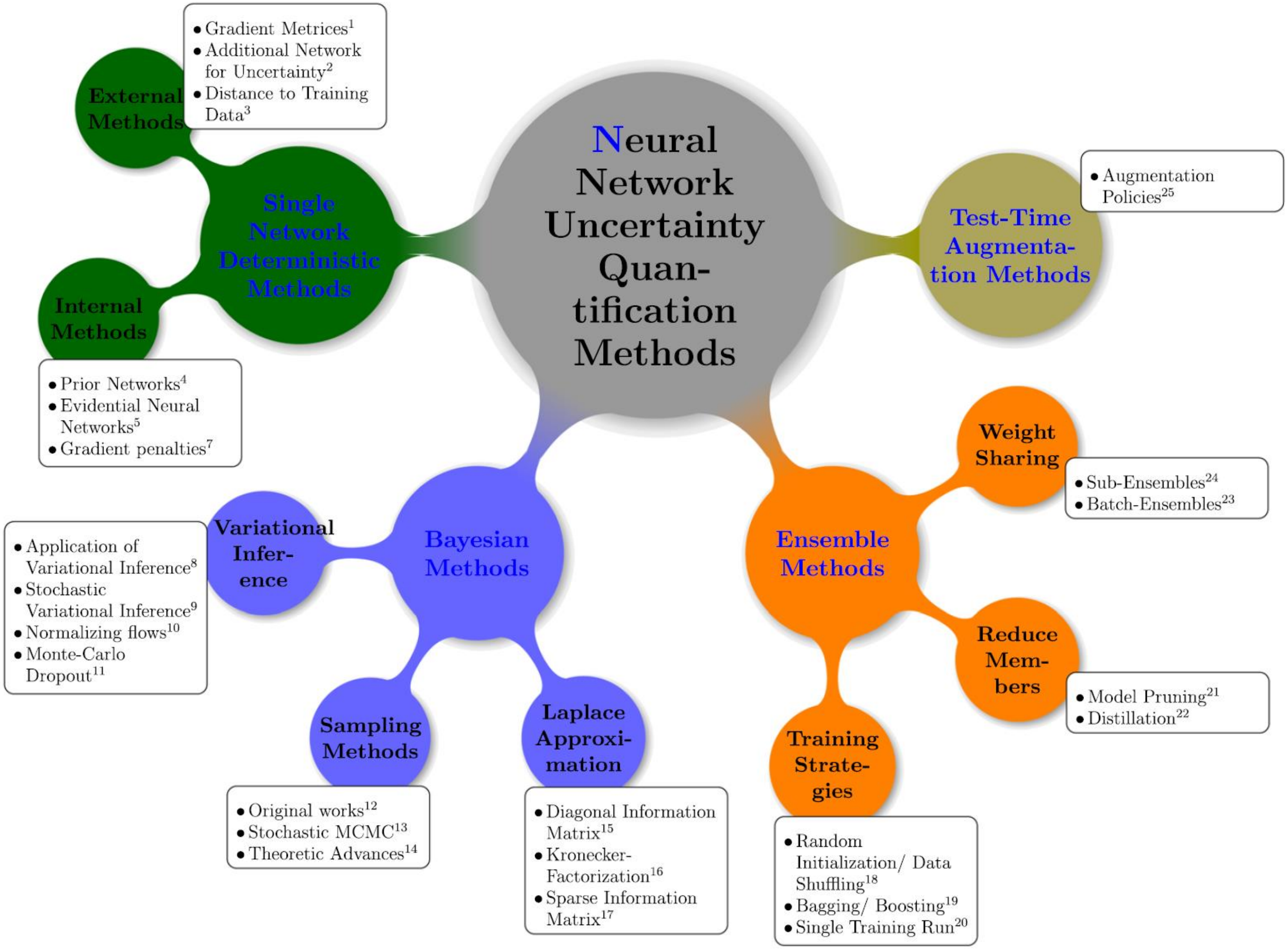
Hüllermeier, Eyke, and Willem Waegeman. "Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods." *Machine Learning* 110 (2021): 457-506.

Where does uncertainty come from?



Gawlikowski, Jakob, et al. "A survey of uncertainty in deep neural networks." *Artificial Intelligence Review* 56.Suppl 1 (2023): 1513-1589.

Approaches



Gawlikowski, Jakob, et al. "A survey of uncertainty in deep neural networks." *Artificial Intelligence Review* 56.Suppl 1 (2023): 1513-1589

Bayesian techniques

- EU can be formulated as a probability distribution over the model parameters.

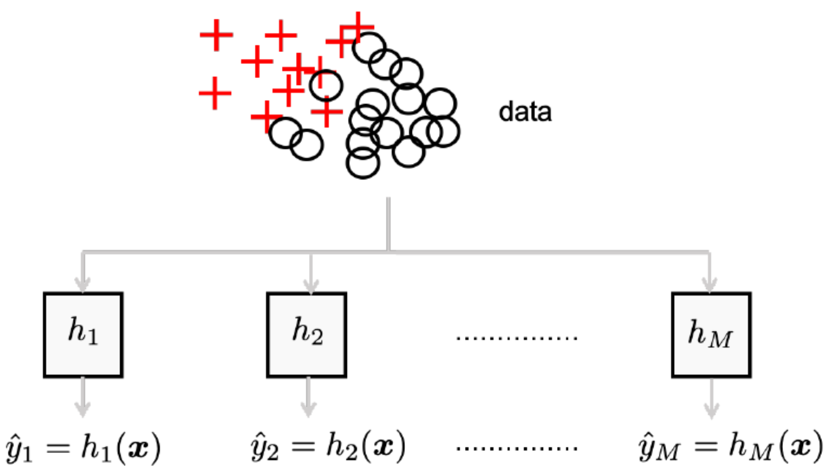
$$p(\omega|X,Y) = \frac{p(Y \mid X,\omega))p(\omega)}{p(Y|X)} = \frac{p(Y \mid X,\omega))p(\omega)}{\boxed{\int p(Y|X,\omega)p(\omega)d\omega}}$$ too costly

$$p(y^*|x^*,X,Y) = \int p(y^*|x^*,\omega)p(\omega|X,Y)d\omega$$

- Use **variational distribution** to approximate
- Use **sampling** to reconstruct.

Ensemble techniques

- Bayesian methods focus on sampling different parameters to represent uncertainty.
- Ensemble methods focus on training different models to represent uncertainty.



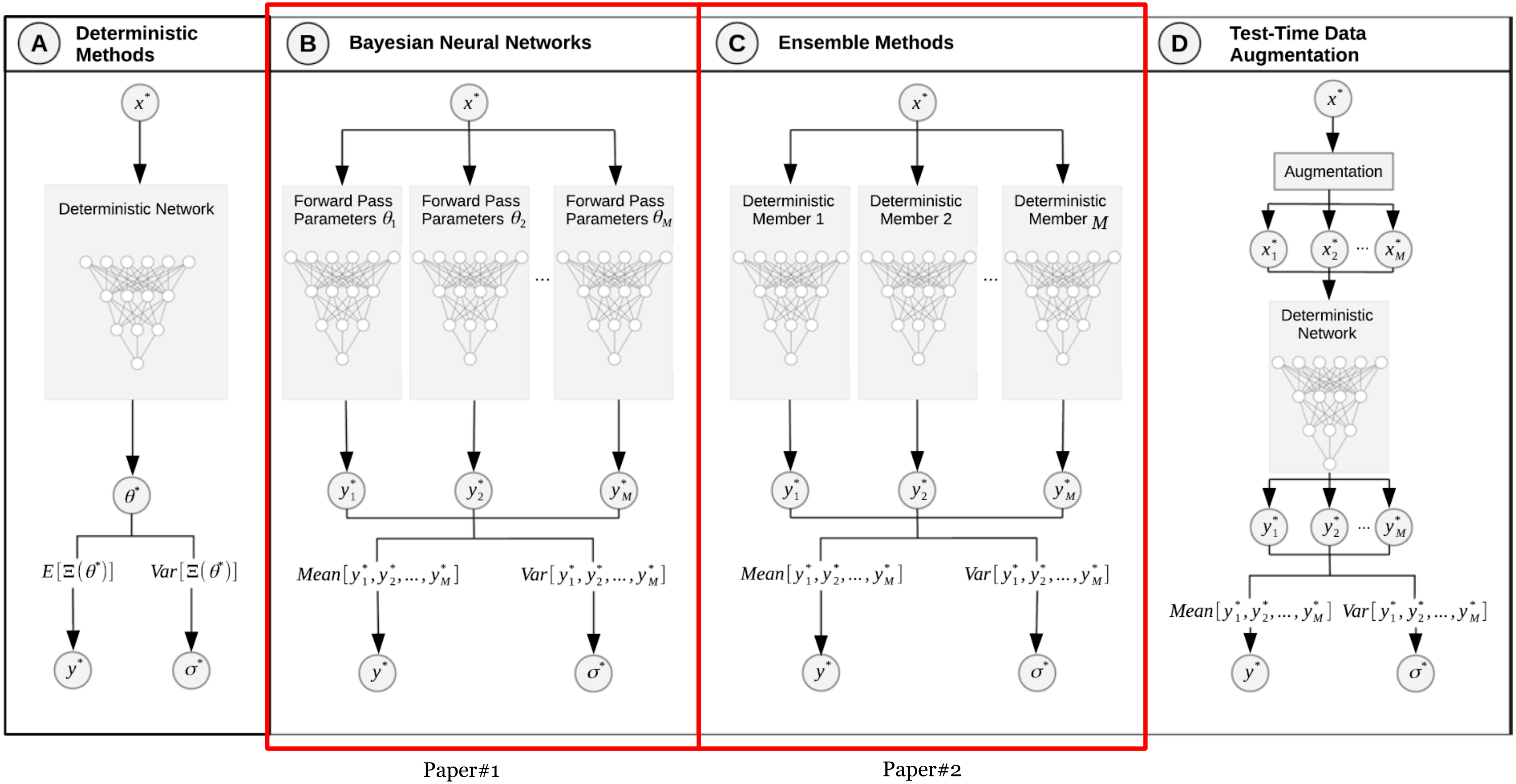
	y_1	y_2	$\dots$	y_K	entropy
$h_1(\boldsymbol{x})$	$p_{1,1}$	$p_{1,2}$	$\dots$	$p_{1,K}$	s_1
$h_2(\boldsymbol{x})$	$p_{2,1}$	$p_{2,2}$	$\dots$	$p_{2,K}$	s_2
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$h_M(\boldsymbol{x})$	$p_{M,1}$	$p_{M,2}$	$\dots$	$p_{M,K}$	s_M
h	p_1	p_2	$\dots$	p_K	$s \mid \bar{s}$

$\mathrm{U}(\boldsymbol{x}) = s =$ entropy of average probabilities

$\mathrm{AU}(\boldsymbol{x}) = \bar{s} =$ average of entropies

$\mathrm{EU}(\boldsymbol{x}) = \mathrm{U}(\boldsymbol{x}) - \mathrm{AU}(\boldsymbol{x})$

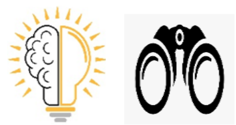
Background and Motivations



Gawlikowski, Jakob, et al. "A survey of uncertainty in deep neural networks." *Artificial Intelligence Review* 56.Suppl 1 (2023): 1513-1589

Roadmap

Background

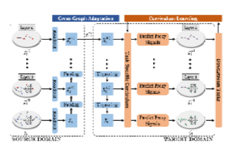


Motivation Overview

I. Background and Motivations

- ➢ What is uncertainty?
- ➢ Why is is important
- ➢ Existing approaches

Paper #1



II. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning

Paper #2

III. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles

Conclusion

IV. Conclusion

Motivations

Dropout as a Bayesian Approximation:
Representing Model Uncertainty in Deep Learning

Yarin Gal YG279@CAM.AC.UK
Zoubin Ghahramani ZG201@CAM.AC.UK
University of Cambridge

- Research Questions
  - How can we **represent model uncertainty** without sacrificing either **computational efficiency** or **accuracy**?
- Problem Definition
  - **Given** a neural network with arbitrary depth and non-linearities.
  - **Find** whether dropout applied before each weight layer is **mathematically equivalent** to an approximation to the probabilistic deep Gaussian process.

Preliminaries

- Notations
  - $\hat{y}$ is the output of a NN model with L layers and loss function $E(y, \hat{y})$
  - W_i is the weight matrices of dimensions $K_i \times K_{i-1}$
  - b_i is the bias vectors of dimension K_i for each layer
  - $L_{dropout} \coloneqq \frac{1}{N} \sum_{i=1}^{N} E(y_i, \hat{y_i}) + \lambda \sum_{i=1}^{L} (\|W_i\|_2^2 + \|b_i\|_2^2)$
  - Sample binary variables which take value 1 with probability p_i for layer i
  - $\omega = \{W_i\}_{i=1}^{L}$
  - Each row of W_i distribute according to the $p(\omega)$
  - m_i is vector of dimension K_i for each gaussian process layer.

Theorem Proof

- Bayesian posterior

$$p(\mathbf{y}|\mathbf{x},\mathbf{X},\mathbf{Y}) = \int p(\mathbf{y}|\mathbf{x},\boldsymbol{\omega})p(\boldsymbol{\omega}|\mathbf{X},\mathbf{Y})\mathrm{d}\boldsymbol{\omega}$$

- First term can be written:

$$p(\mathbf{y}|\mathbf{x},\boldsymbol{\omega}) = \mathcal{N}\big(\mathbf{y}; \widehat{\mathbf{y}}(\mathbf{x},\boldsymbol{\omega}), \tau^{-1}\mathbf{I}_D\big)$$

- Furthermore, the prediction can be written:

$$\begin{aligned}\widehat{\mathbf{y}}\big(\mathbf{x},\boldsymbol{\omega} = \{\mathbf{W}_1,...,\mathbf{W}_L\}\big) \\ = \sqrt{\frac{1}{K_L}}\mathbf{W}_L\sigma\Big(...\sqrt{\frac{1}{K_1}}\mathbf{W}_2\sigma\big(\mathbf{W}_1\mathbf{x}+\mathbf{m}_1\big)...\Big)\end{aligned}$$

Theorem Proof

- Second term is intractable

$$p(\mathbf{y}|\mathbf{x},\mathbf{X},\mathbf{Y}) = \int p(\mathbf{y}|\mathbf{x},\boldsymbol{\omega})\boxed{p(\boldsymbol{\omega}|\mathbf{X},\mathbf{Y})}^{?}\mathrm{d}\boldsymbol{\omega}$$

- **Use another distribution to approximate it.**
- $q(\boldsymbol{\omega})$ is a distribution whose columns are randomly set to zero:

$$\mathbf{W}_i = \mathbf{M}_i \cdot \mathrm{diag}([\mathbf{z}_{i,j}]_{j=1}^{K_i})$$

$$\mathbf{z}_{i,j} \sim \mathrm{Bernoulli}(p_i) \text{ for } i = 1, ..., L,\ j = 1, ..., K_{i-1}$$

  - p_i is Bernoulli probability
  - M_i is the matrices of variational parameters.
  - The binary variable $z_{i,j} = 0$ corresponds then to unit j in layer $i - 1$ being dropped out as an input to layer i.

Theorem Proof

- Now we get $q(\omega)$ to approximate posterior $p(\omega|X,Y)$
- We want $q(\omega)$ **close** to $p(\omega|X,Y)$, hence **KL divergence** is used to optimize $q(\omega)$:

$$KL(q(\omega)||p(\omega|X,Y) = \int q(\omega)\log\frac{q(\omega)}{p(\omega|X,Y)}\,d\omega$$

- Minimizing the KL divergence is equivalent to maximizing the **log evidence lower bound**:

$$-\int q(\omega)\log p(Y|X,\omega)d\omega + KL(q(\omega)||p(\omega))$$

Bishop, C. M. (2006). Pattern Recognition and Machine Learning (Information Science and Statistics). Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Theorem Proof

- How to minimize

$$-\int q(\omega)\log p(Y|X,\omega)d\omega + KL(q(\omega)||p(\omega))$$

- First term can be rewritten as:

$$-\sum_{n=1}^{N}\int q(\omega)\log p(y_n|x_n,\omega)d\omega$$

- Completely computation is costly, so we use MC integration to approximate it.

$$\widehat{\omega_n} \sim q(\omega)$$

$$\sim -\sum_{n=1}^{N} logp(y_n|x_n,\widehat{\omega_n})$$

Theorem Proof

- Second term can be rewritten as:

$$-\sum_{i=1}^{L}(\frac{p_i l^2}{2}\|M_i\|_2^2 + \frac{l^2}{2}\|m_i\|_2^2)$$

- model precision τ
- length-scale l, $p_l(\omega) = N(\omega; 0, l^{-2}I_K)$, decouples precision from weight decays λ

- Combined with first term and second term, we get:
  - $L_{GP-MC} \propto \frac{1}{N}\sum_{n=1}^{N}\frac{-logp(y_n|x_n,\widehat{\omega_n})}{\tau} + \sum_{i=1}^{L}\left(\frac{p_i l^2}{2\tau N}\|M_i\|_2^2 + \frac{l^2}{2\tau N}\|m_i\|_2^2\right)$
  - $L_{dropout} := \frac{1}{N}\sum_{i=1}^{N} E(y_i, \hat{y_i}) + \lambda\sum_{i=1}^{L}(\|W_i\|_2^2 + \|b_i\|_2^2)$
- Thus, by setting $E\left(y_n, \hat{y}(x_n, \widehat{\omega_n})\right) = \frac{-logp(y_n|x_n,\widehat{\omega_n})}{\tau}$, we can recover original loss function.
- Finish proof.

Obtaining Model Uncertainty

- Recall the covariance function:

$$Var(X) = E(X^2) - E(X)^2$$

- Since we get the proof, then the expectation can be written as:

$$\mathbb{E}_{q(\mathbf{y}^*|\mathbf{x}^*)}(\mathbf{y}^*) \approx \frac{1}{T}\sum_{t=1}^{T} \widehat{\mathbf{y}}^*(\mathbf{x}^*, \mathbf{W}_1^t, ..., \mathbf{W}_L^t)$$

$$\mathbb{E}_{q(\mathbf{y}^*|\mathbf{x}^*)}\big((\mathbf{y}^*)^T(\mathbf{y}^*)\big) \approx \tau^{-1}\mathbf{I}_D + \frac{1}{T}\sum_{t=1}^{T} \widehat{\mathbf{y}}^*(\mathbf{x}^*, \mathbf{W}_1^t, ..., \mathbf{W}_L^t)^T \widehat{\mathbf{y}}^*(\mathbf{x}^*, \mathbf{W}_1^t, ..., \mathbf{W}_L^t)$$

$$\mathrm{Var}_{q(\mathbf{y}^*|\mathbf{x}^*)}\big(\mathbf{y}^*\big) \approx \tau^{-1}\mathbf{I}_D + \frac{1}{T}\sum_{t=1}^{T} \widehat{\mathbf{y}}^*(\mathbf{x}^*, \mathbf{W}_1^t, ..., \mathbf{W}_L^t)^T \widehat{\mathbf{y}}^*(\mathbf{x}^*, \mathbf{W}_1^t, ..., \mathbf{W}_L^t) - \mathbb{E}_{q(\mathbf{y}^*|\mathbf{x}^*)}(\mathbf{y}^*)^T \mathbb{E}_{q(\mathbf{y}^*|\mathbf{x}^*)}(\mathbf{y}^*)$$

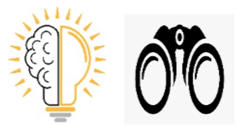
Predictive performance

- Dropout methods can achieve a competitive accuracy comparing to VI and PBP baselines

			Avg. Test RMSE and Std. Errors			Avg. Test LL and Std. Errors		
Dataset	*N*	*Q*	**VI**	**PBP**	**Dropout**	**VI**	**PBP**	**Dropout**
Boston Housing	506	13	4.32 ±0.29	3.01 ±0.18	**2.97 ±0.19**	-2.90 ±0.07	-2.57 ±0.09	**-2.46 ±0.06**
Concrete Strength	1,030	8	7.19 ±0.12	5.67 ±0.09	**5.23 ±0.12**	-3.39 ±0.02	-3.16 ±0.02	**-3.04 ±0.02**
Energy Efficiency	768	8	2.65 ±0.08	1.80 ±0.05	**1.66 ±0.04**	-2.39 ±0.03	-2.04 ±0.02	**-1.99 ±0.02**
Kin8nm	8,192	8	**0.10 ±0.00**	**0.10 ±0.00**	**0.10 ±0.00**	0.90 ±0.01	0.90 ±0.01	**0.95 ±0.01**
Naval Propulsion	11,934	16	**0.01 ±0.00**	**0.01 ±0.00**	**0.01 ±0.00**	3.73 ±0.12	3.73 ±0.01	**3.80 ±0.01**
Power Plant	9,568	4	4.33 ±0.04	4.12 ±0.03	**4.02 ±0.04**	-2.89 ±0.01	-2.84 ±0.01	**-2.80 ±0.01**
Protein Structure	45,730	9	4.84 ±0.03	4.73 ±0.01	**4.36 ±0.01**	-2.99 ±0.01	-2.97 ±0.00	**-2.89 ±0.00**
Wine Quality Red	1,599	11	0.65 ±0.01	0.64 ±0.01	**0.62 ±0.01**	-0.98 ±0.01	-0.97 ±0.01	**-0.93 ±0.01**
Yacht Hydrodynamics	308	6	6.89 ±0.67	**1.02 ±0.05**	1.11 ±0.09	-3.43 ±0.16	-1.63 ±0.02	**-1.55 ±0.03**
Year Prediction MSD	515,345	90	9.034 ±NA	8.879 ±NA	**8.849 ±NA**	-3.622 ±NA	-3.603 ±NA	**-3.588 ±NA**

Roadmap

Background

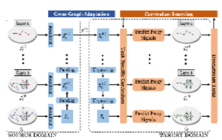


Motivation Overview

I. Background and Motivations

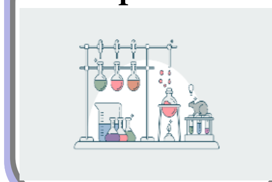
- ➢ What is uncertainty?
- ➢ Why is is important
- ➢ Existing approaches

Paper #1



II. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning

Paper #2



III. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles

Conclusion

IV. Conclusion

Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles

Balaji Lakshminarayanan **Alexander Pritzel** **Charles Blundell**
DeepMind
{balajiln,apritzel,cblundell}@google.com

- Research Questions
  - Dropout can be regarded as a form of ensemble
  - Can ensemble be an **alternative** for estimating uncertainty?
- Problem Definition
  - **Given** ensemble framework
  - **Find** whether **ensemble and adversarial training** can be used to estimate predictive uncertainty

Recipe

- This paper mainly focus on evaluating an ensemble framework's capability on representing uncertainty
- Essential steps
  - Choose a proper scoring rule as the training criterion
    - Likelihood/Softmax cross entropy
  - Use adversarial training argument samples to smooth the predictive distributions
    - $x' = x + \epsilon \text{sign}(\nabla x\, l(\theta, x, y))$
  - Ensemble to calculate uncertainty
    - No bagging
    - Random initialization of different models
    - Minimize $l(\theta_m, x_{n_m}, y_{n_m}) + l(\theta_m, x'_{n_m}, y_{n_m})$ for each learner

Regression on real world datasets

- Ensemble is competitive with PBP(probabilistic backpropagation) and MC-dropout in terms of NLL
- But is slightly worse in terms of RMSE
- No significant improvement compared to MC-dropout

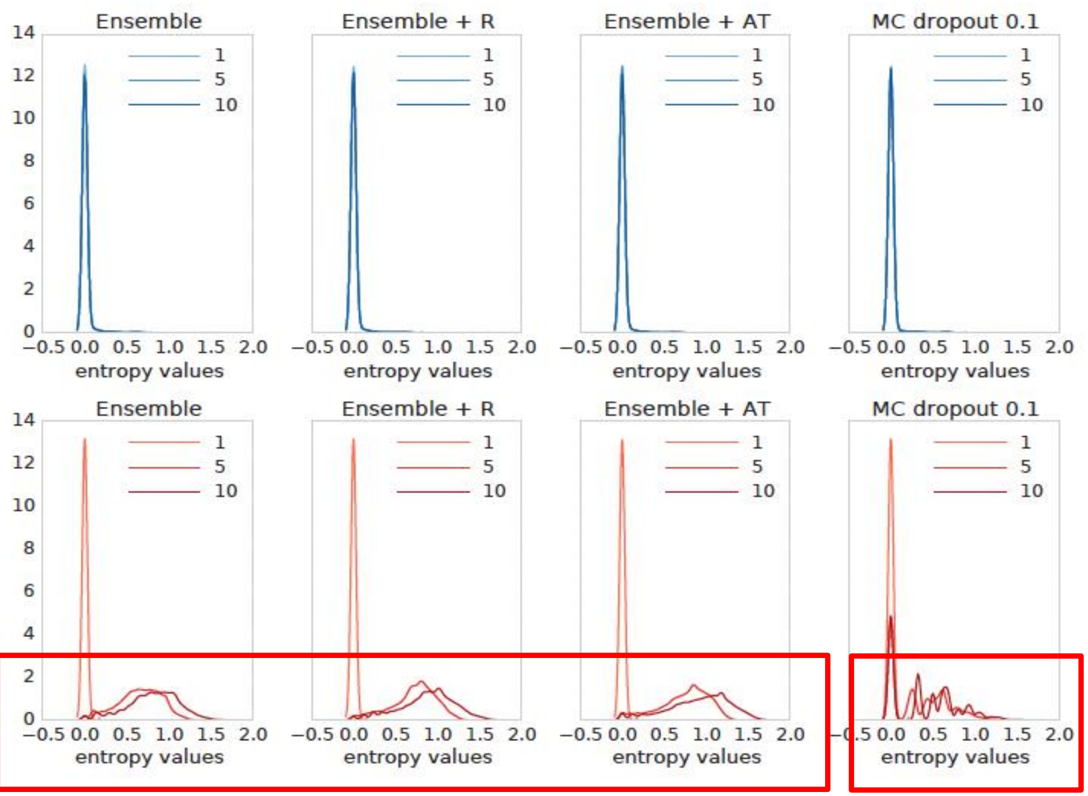
Datasets	RMSE			NLL		
	PBP	MC-dropout	Deep Ensembles	PBP	MC-dropout	Deep Ensembles
Boston housing	**3.01 ± 0.18**	**2.97 ± 0.85**	**3.28 ± 1.00**	**2.57 ± 0.09**	**2.46 ± 0.25**	**2.41 ± 0.25**
Concrete	**5.67 ± 0.09**	**5.23 ± 0.53**	**6.03 ± 0.58**	**3.16 ± 0.02**	**3.04 ± 0.09**	**3.06 ± 0.18**
Energy	**1.80 ± 0.05**	**1.66 ± 0.19**	**2.09 ± 0.29**	2.04 ± 0.02	1.99 ± 0.09	**1.38 ± 0.22**
Kin8nm	0.10 ± 0.00	0.10 ± 0.00	**0.09 ± 0.00**	-0.90 ± 0.01	-0.95 ± 0.03	**-1.20 ± 0.02**
Naval propulsion plant	0.01 ± 0.00	0.01 ± 0.00	**0.00 ± 0.00**	-3.73 ± 0.01	-3.80 ± 0.05	**-5.63 ± 0.05**
Power plant	**4.12 ± 0.03**	**4.02 ± 0.18**	**4.11 ± 0.17**	2.84 ± 0.01	**2.80 ± 0.05**	**2.79 ± 0.04**
Protein	4.73 ± 0.01	**4.36 ± 0.04**	4.71 ± 0.06	2.97 ± 0.00	2.89 ± 0.01	**2.83 ± 0.02**
Wine	**0.64 ± 0.01**	**0.62 ± 0.04**	**0.64 ± 0.04**	0.97 ± 0.01	**0.93 ± 0.06**	**0.94 ± 0.12**
Yacht	**1.02 ± 0.05**	**1.11 ± 0.38**	**1.58 ± 0.48**	1.63 ± 0.02	1.55 ± 0.12	**1.18 ± 0.21**
Year Prediction MSD	8.88 ± NA	**8.85 ± NA**	8.89 ± NA	3.60 ± NA	3.59 ± NA	**3.35 ± NA**

Out of distribution samples evaluation

- MC-dropout seems to give high confidence predictions for unknown test examples.
- Adversarial training has no effect on unknown samples.

known

unknown



not much difference when applying adversarial training

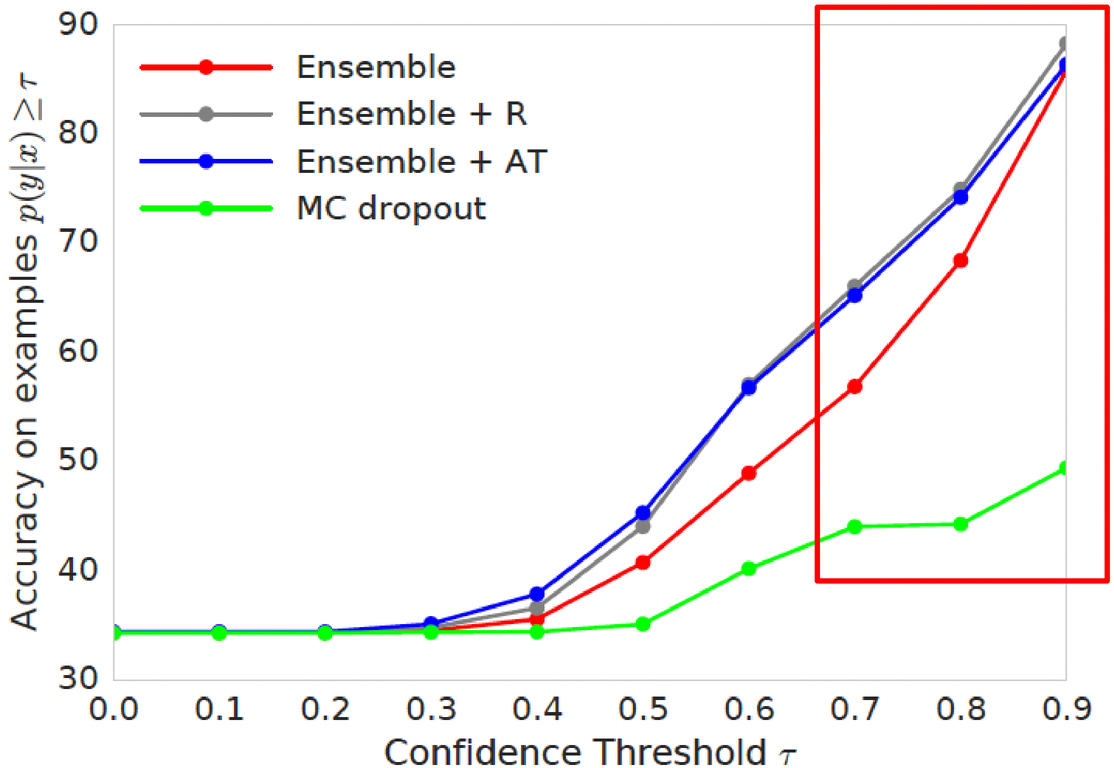
not much difference when model num increases

The images in NotMNIST dataset have the same size as MNIST, however the labels are alphabets instead of digits.

(a) MNIST-NotMNIST

Uncertainty vs Accuracy

- Ensemble achieves better performance in high confidence samples.



The model is evaluated only on cases where the model's confidence is above a user-specified threshold.

$$\hat{y} = \arg\max_k p(y = k|\mathbf{x})$$

$$p(y = \hat{y}|\mathbf{x}) = \max_k p(y = k|\mathbf{x}).$$

MC dropout performs worse when in high confidence threshold

Roadmap

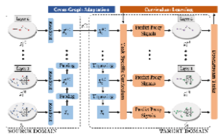
Background

Motivation Overview

I. Background and Motivations

- ➢ What is uncertainty?
- ➢ Why is is important
- ➢ Existing approaches

Paper #1



II. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning

Paper #2

III. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles

Conclusion

IV. Conclusion

Conclusion

Approach	Pros	Cons
Dropout	• **Solid theoretical analysis.** • Detailed empirical analysis • **Unbiased from the model's predictions**	• **Time consuming comparing to ensemble methods** • Uncertainty depends on training data and prior distributions
Ensemble	• Improvement in accuracy and representation of uncertainty • **Practicality and intuitive** • Detailed empirical analysis	• Originally not introduced to explicitly handle uncertainties


Q
&
a
!