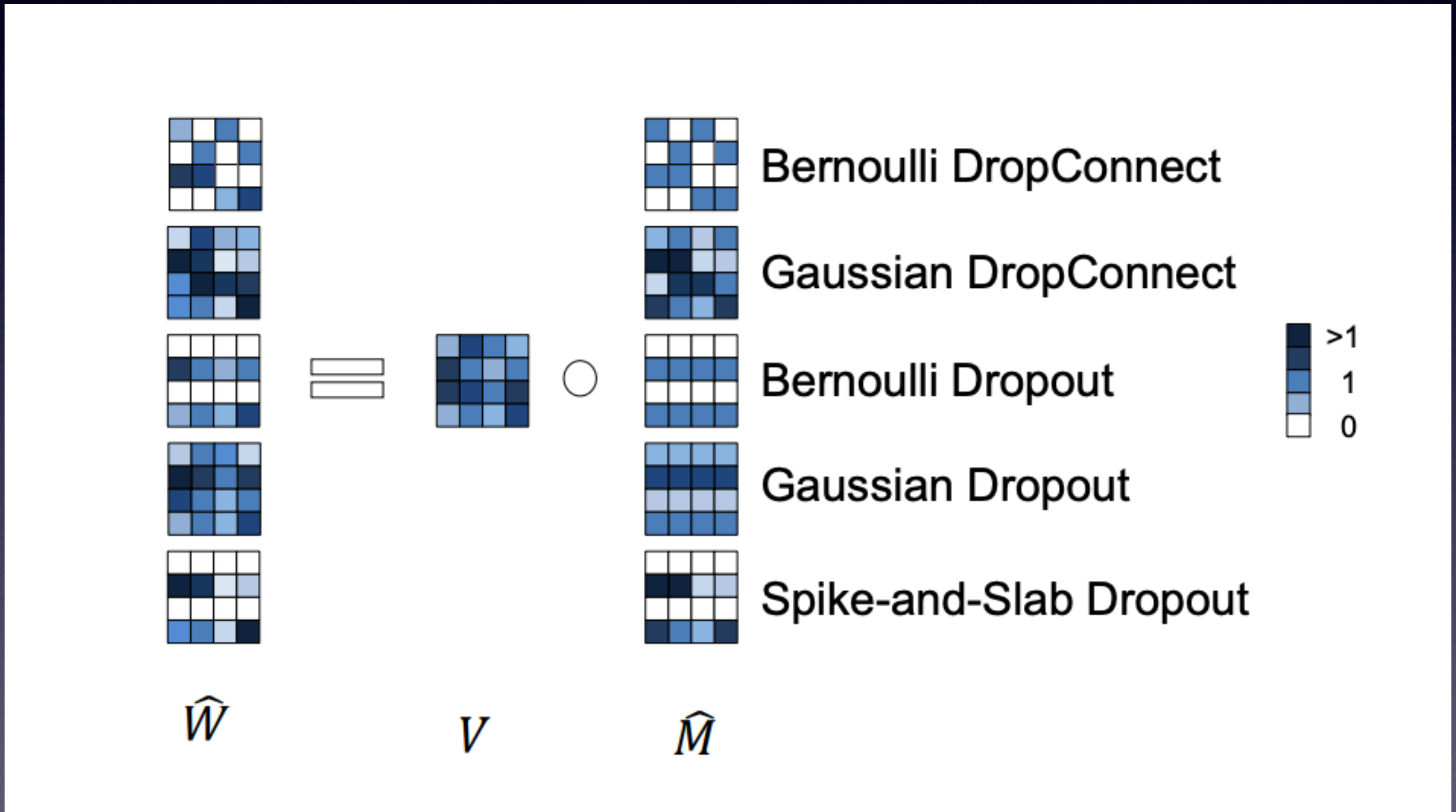
Robustly representing uncertainty through sampling in deep neural networks



Fabio Peruzzo

Overview

1. Uncertainty estimation and neural networks
2. A look back at variational approximation in Bayesian NN methods
3. “Robustly representing uncertainty through sampling”
4. DropConnect beats DropOut

Uncertainty estimation and neural networks

aleatoric uncertainty:
uncertainty present in the training data
(estimated e.g. through softmax output)
It cannot be reduced by collecting more data

epistemic uncertainty:
parameter uncertainty, coming from training process

Uncertainty estimation and neural networks

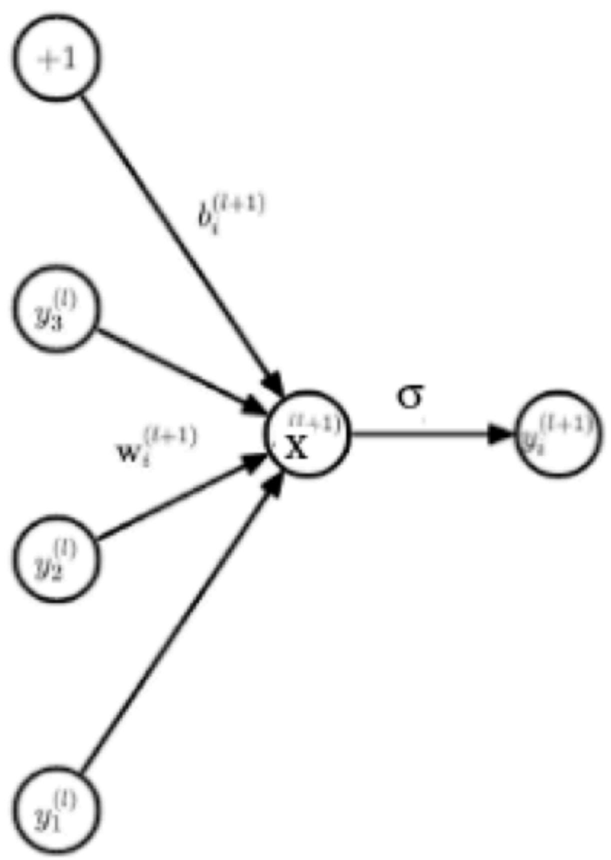
aleatoric uncertainty:
uncertainty present in the training data
(estimated e.g. through softmax output)
It cannot be reduced by collecting more data

epistemic uncertainty:
parameter uncertainty, coming from training process

=> Bayesian DNNs attempt to learn a distribution over their parameters thereby allowing for the computation of epistemic uncertainty

However, ideal Bayesian methods do not scale well due to the difficulty in computing, so **we need to rely on approximate methods**

Epistemic uncertainty: ideal case



(a) Standard network

$$x^{(l+1)} = W^{(l+1)} y^{(l)} + b^{(l+1)}$$

$$y^{(l+1)} = \sigma\left(x^{(l+1)}\right)$$

How to estimate uncertainty coming from the training process?

We would need to re-train the model several (hundreds of) times.

Approximate methods for epistemic uncertainty

Among the most famous approaches for approximate Bayesian inference:

1. **Laplace approximation:**
   David JC MacKay. A practical bayesian framework for backpropagation networks. Neural computation, 4(3):448–472, 1992.

2. **Markov Chain Monte Carlo**
   Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In Proceedings of the 28th International Conference on Machine Learning (ICML-11), pages 681–688, 2011.

3. **Variational approaches**
   Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Insights and applications. In Deep Learning Workshop, ICML, 2015.

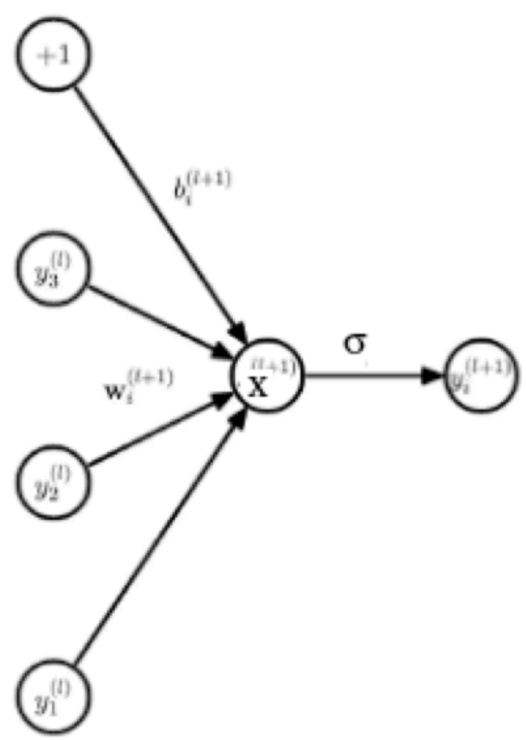
Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning

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

Introduces a theoretical framework that links
dropout training <=> deep Gaussian processes
through Bayesian inference

This paper proved how and with which assumptions dropout at inference can be used for uncertainty estimation

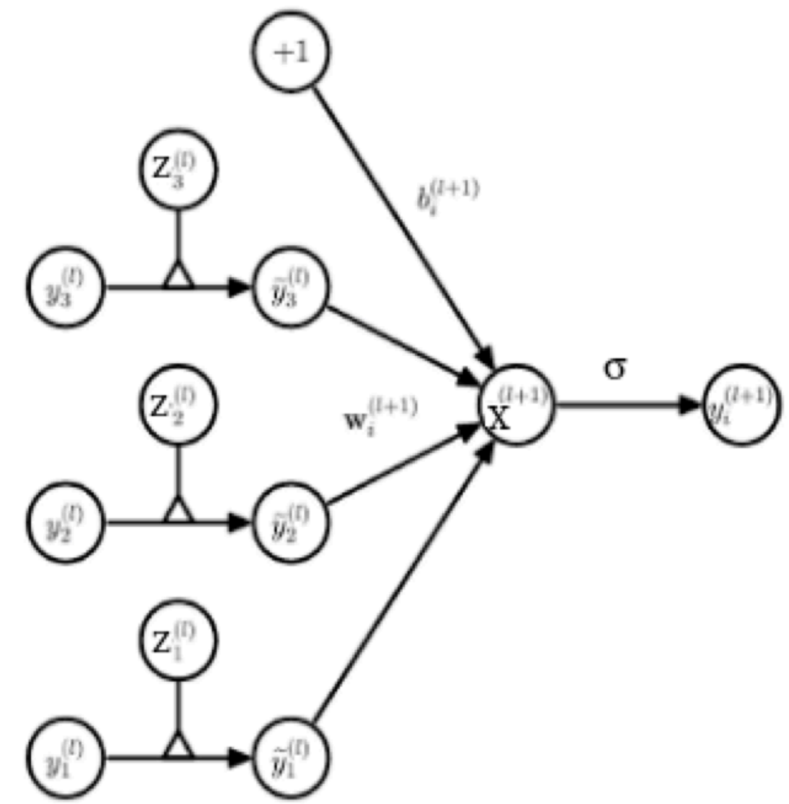
Dropout: model sampling



(a) Standard network

$$x^{(l+1)} = W^{(l+1)} y^{(l)} + b^{(l+1)}$$

$$y^{(l+1)} = \sigma(x^{(l+1)})$$

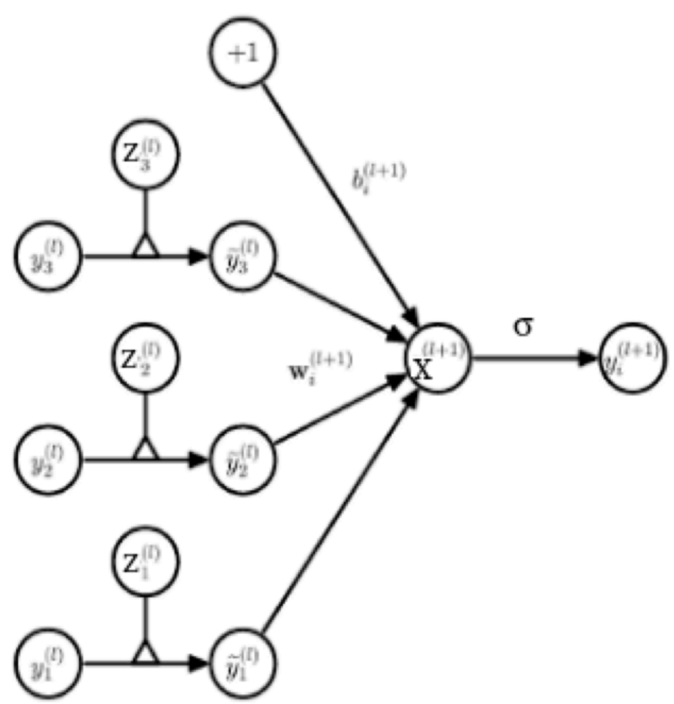


(b) Dropout network

$$z^{(l)} = (z_1, z_2, \ldots) \text{ with } z_i \sim Bernoulli(p)$$

$$\tilde{y}^{(l)} = z^{(l)} \odot y^{(l)} = (z_1^{(l)} y_1^{(l)}, z_2^{(l)} y_2^{(l)}, \ldots)$$

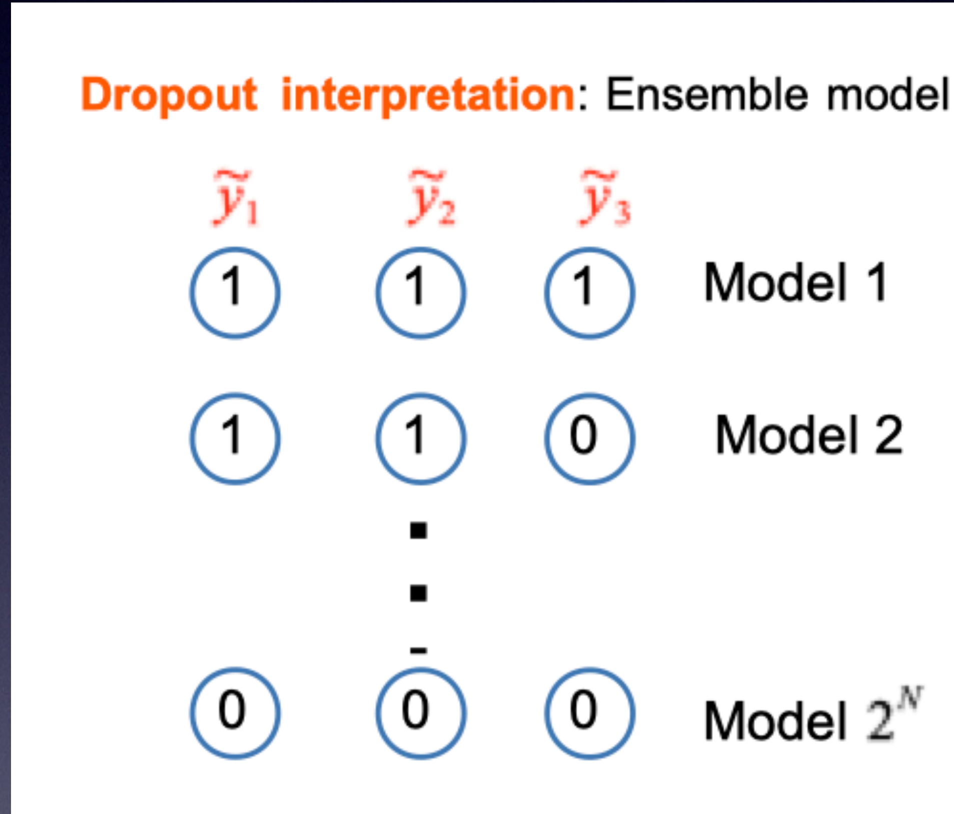
Dropout: model sampling



(b) Dropout network

$$z^{(l)} = (z_1, z_2, \ldots) \text{ with } z_i \sim Bernoulli(p)$$

$$\widetilde{y}^{(l)} = z^{(l)} \odot y^{(l)} = (z_1^{(l)} y_1^{(l)}, z_2^{(l)} y_2^{(l)}, \ldots)$$



Variational Methods for Bayesian NNs

Predicting y in Bayesian Inference:

$$P(\hat{y}/\hat{x}, X, Y) = \int P(\hat{y}/\hat{x}, W_1, W_2, b) \boxed{P(W_1, W_2, b/X, Y)} dW_1 dW_2 db$$

Problem: estimating the posterior

Variational Methods for Bayesian NNs

Predicting y in Bayesian Inference:

$$P(\hat{y}/\hat{x},X,Y)=\int P(\hat{y}/\hat{x},W_1,W_2,b)\boxed{P(W_1,W_2,b/X,Y)}dW_1dW_2db$$

Problem: estimating the posterior

The variational approximation

$$P(W_1,W_2b/X,Y)\sim q_M(W_1,W_2,b)=q_{M_1}(W_1)q_{M_2}(W_2)q_m(b)$$

Dropout and Variational Methods

If we take a Deep Gaussian Process and

$$q_M(W)=\prod_\alpha q_{m_\alpha}(w_\alpha) \text{ with } w_\alpha/m_\alpha \text{ the colums of } W/M$$

$$q_{m_\alpha}(w_\alpha)=pN(m_\alpha,\theta^2 I)+(1-p)*N(0,\theta^2 I)$$

$$q(b)=N(m,\theta^2 I)$$

Dropout and Variational Methods

If we take a Deep Gaussian Process and

$$q_M(W)=\prod_{\alpha} q_{m_\alpha}(w_\alpha) \text{ with } w_\alpha / m_\alpha \text{ the colums of } W/M$$

$$q_{m_\alpha}(w_\alpha)=pN(m_\alpha, \theta^2 I)+(1-p)*N(0, \theta^2 I)$$

$$q(b)=N(m, \theta^2 I)$$

and if we train the Bayesian NN to maximise the ELBO

$$ELBO(q_M(W_1, W_2, b))=E_{W_1, W_2, b\sim q_M(W_1, W_2, b)}[\ln P(D/W_1, W_2, b)]-KL(q_M(W_1, W_2, b) \| P(W_1, W_2, b))$$

Likelihood prior

in the limit θ -> 0, the inference becomes what we expect:

$$E_{q_M(y^*/x^*)}(y^*)\simeq\frac{1}{T}\sum_{t=1}^{T} \hat{y}^*(x^*, z_1^t, z_2^t, \ldots)$$

Dropout and Variational Methods

So, a Deep Gaussian Process with

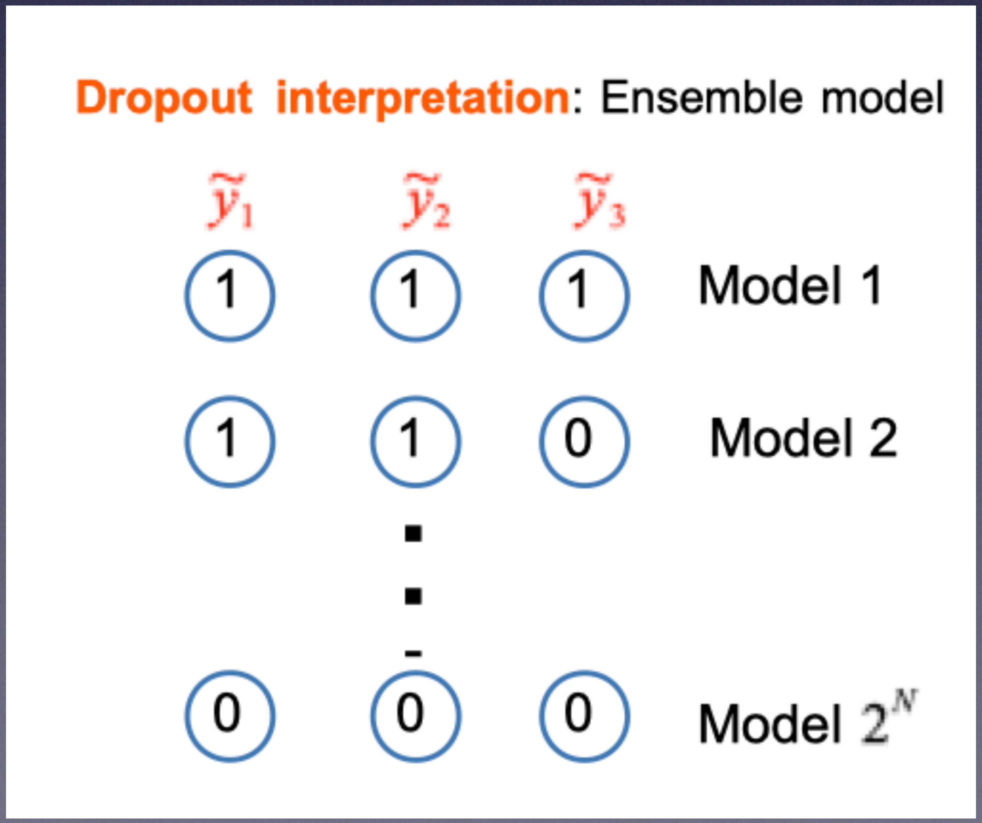
$$q_M(W) = \prod_\alpha q_{m_\alpha}(w_\alpha) \text{ with } w_\alpha / m_\alpha \text{ the colums of } W/M$$

$$q_{m_\alpha}(w_\alpha) = pN(m_\alpha, \theta^2 I) + (1-p) * N(0, \theta^2 I)$$

$$q(b) = N(m, \theta^2 I)$$

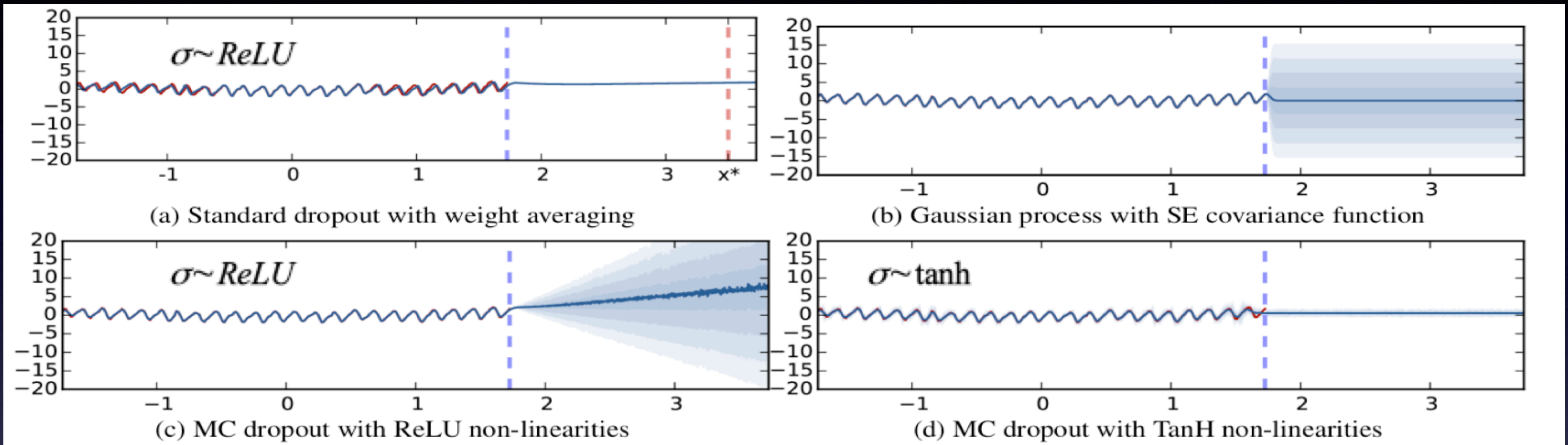
Is equivalent to independently sampling models through dropout

$$E_{q_M(y^*/x^*)}(y^*) \simeq \frac{1}{T} \sum_{t=1}^{T} \hat{y}^*(x^*, z_1^t, z_2^t, \ldots)$$



Experiment: Mauna Loa CO2 concentrations

We can (approximately) infer the uncertainty of the model



(a) Standard dropout with weight averaging
(b) Gaussian process with SE covariance function
(c) MC dropout with ReLU non-linearities
(d) MC dropout with TanH non-linearities

- NN with 4 or 5 hidden layers and 1024 hidden units
- Fig 2b SE = squared exponential
- None of the models captures periodicity
- Strong dependence on activation functions of uncertainty bands
- Seems to imply that ReLU is very unstable -> untrue!

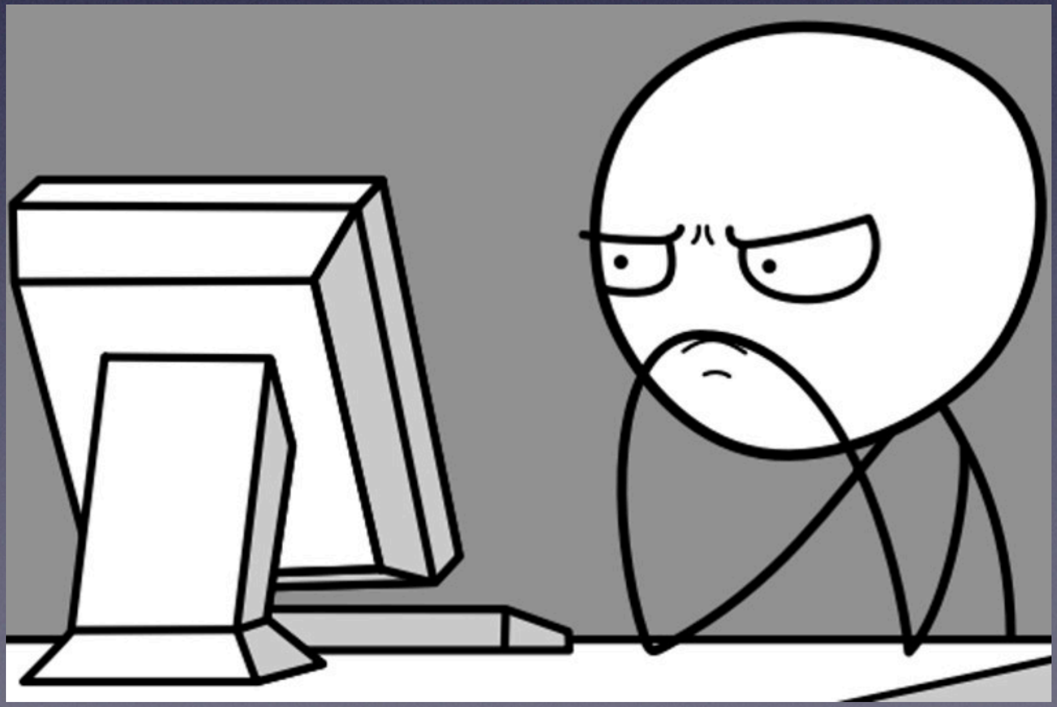
Conclusions on
"Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning"

- Dropout can be used to estimate epistemic uncertainty
- There is a direct connection between DGM and dropout sampling
- This connection can be proved using Variational approximation

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

- Dropout can be used to estimate epistemic uncertainty
- There is a direct connection between DGM and dropout sampling
- This connection can be proved using Variational approximation

However:
- Goodness of such approximation is unclear
- strong dependence on activation function



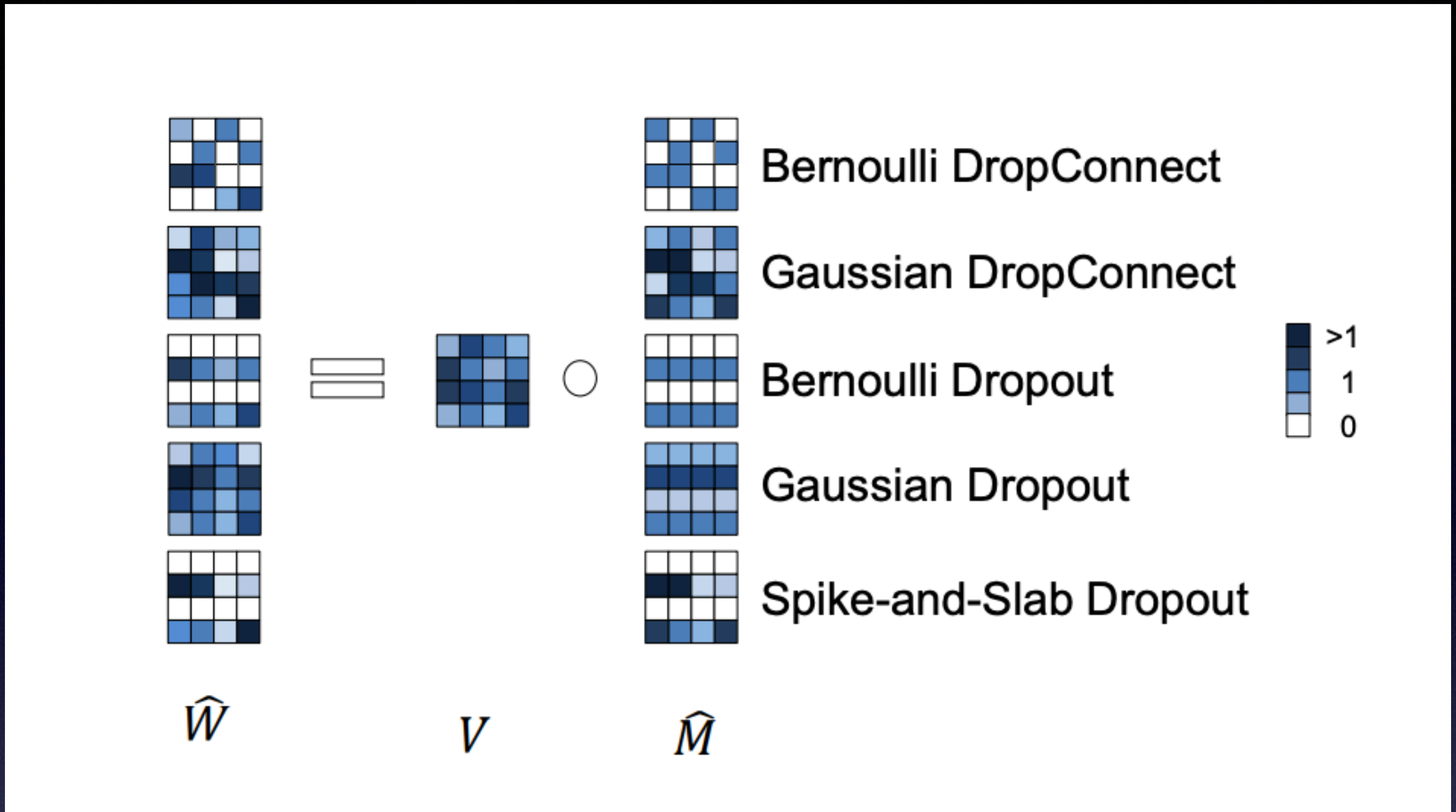
So… is it any good?

Robustly representing uncertainty through sampling in deep neural networks

Patrick McClure
MRC Cognition and Brain Sciences Unit
University of Cambridge
patrick.mcclure@mrc-cbu.cam.ac.uk

Nikolaus Kriegeskorte
Department of Psychology
Columbia University
nk2765@columbia.edu

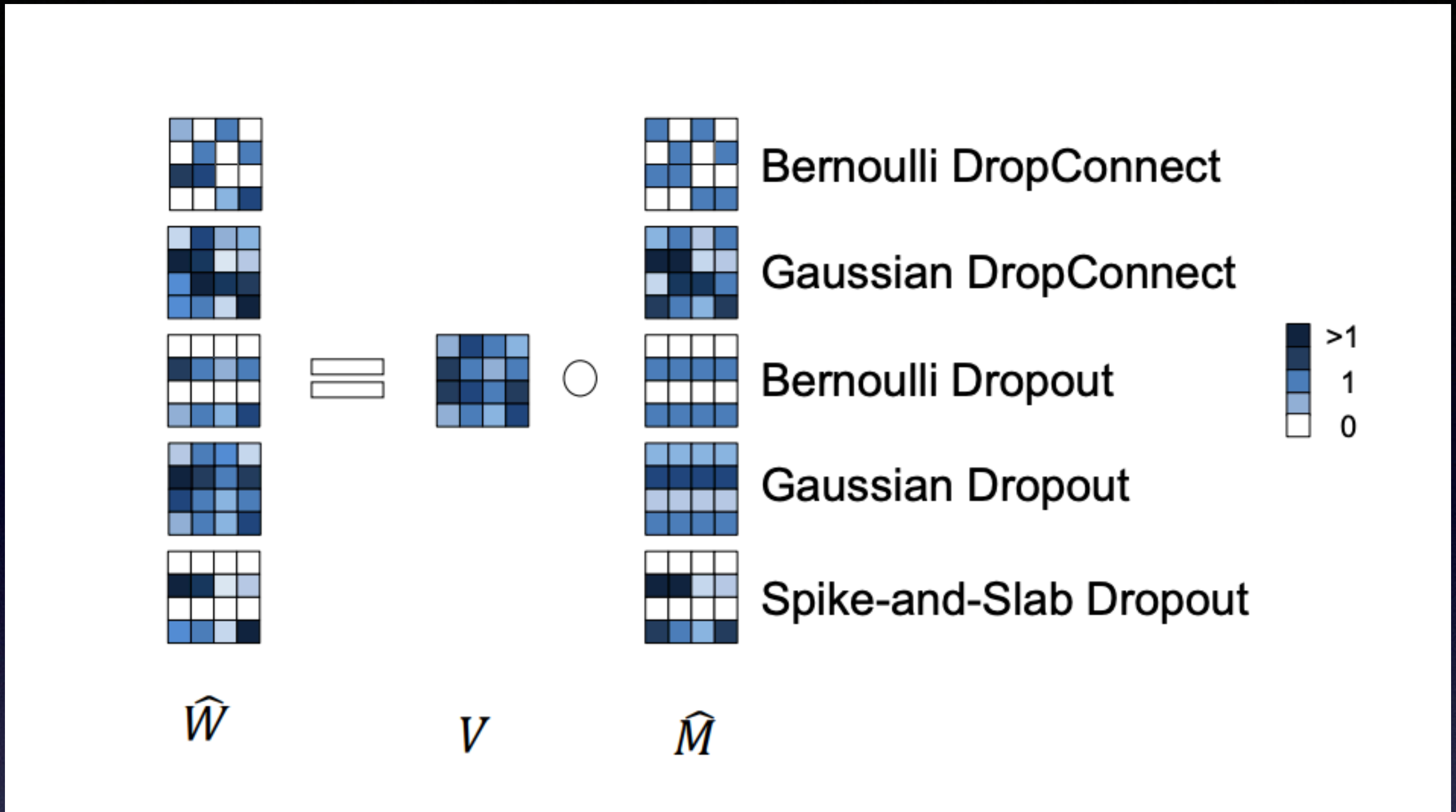
- It explores generalisations of dropout
- Tests on two examples (MNIST and Cifar-10)
- Not a very successful paper (has it been published?)



$$\hat{W} = V \circ \hat{M} \quad where \quad \hat{M} \sim p(M)$$

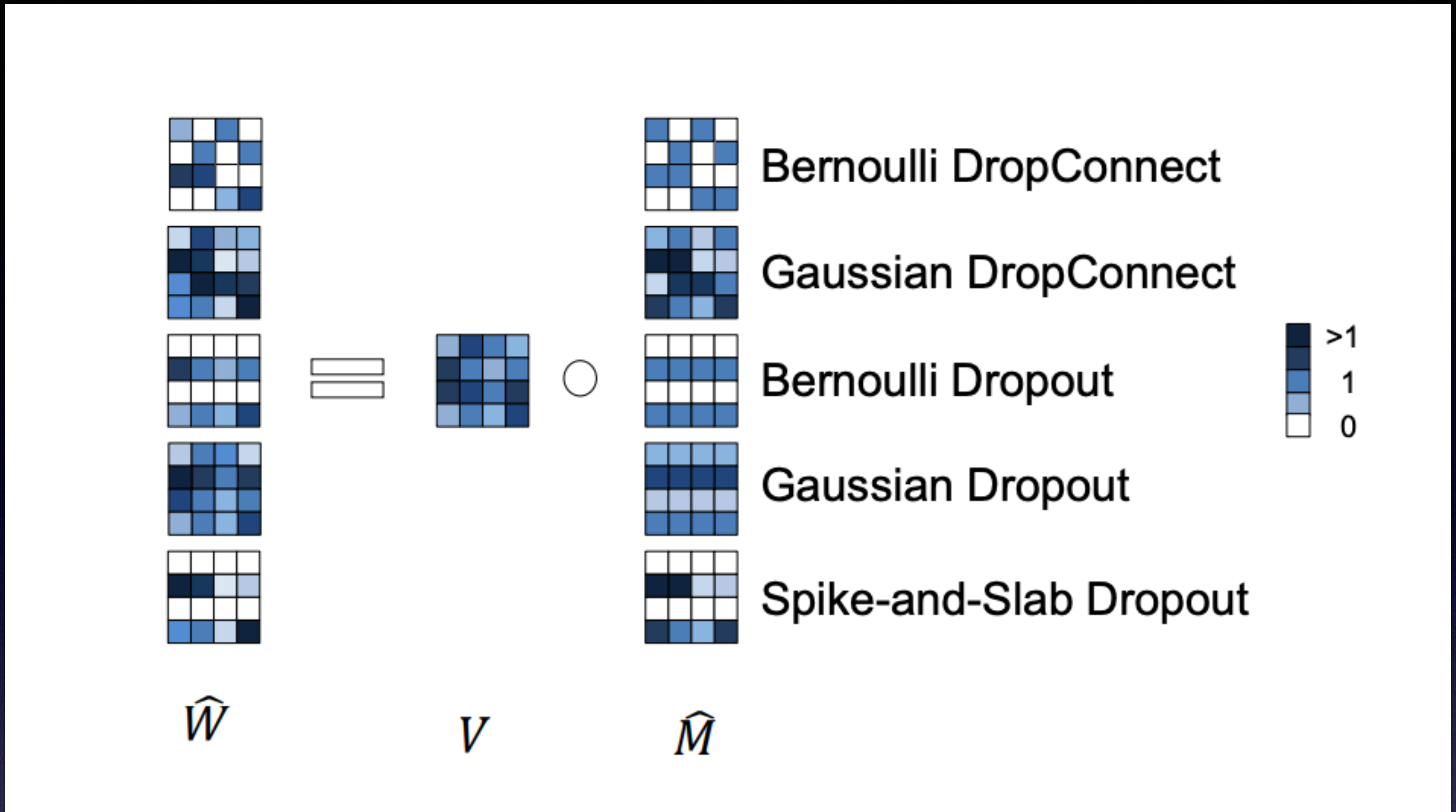
Where

- W, V, M are matrices with **one entry for each connection in the NN**
- W are the sampled weights of the NN
- V are the variational parameters (the “unmodified” weights)
- M is a mask which samples a perturbation to the model



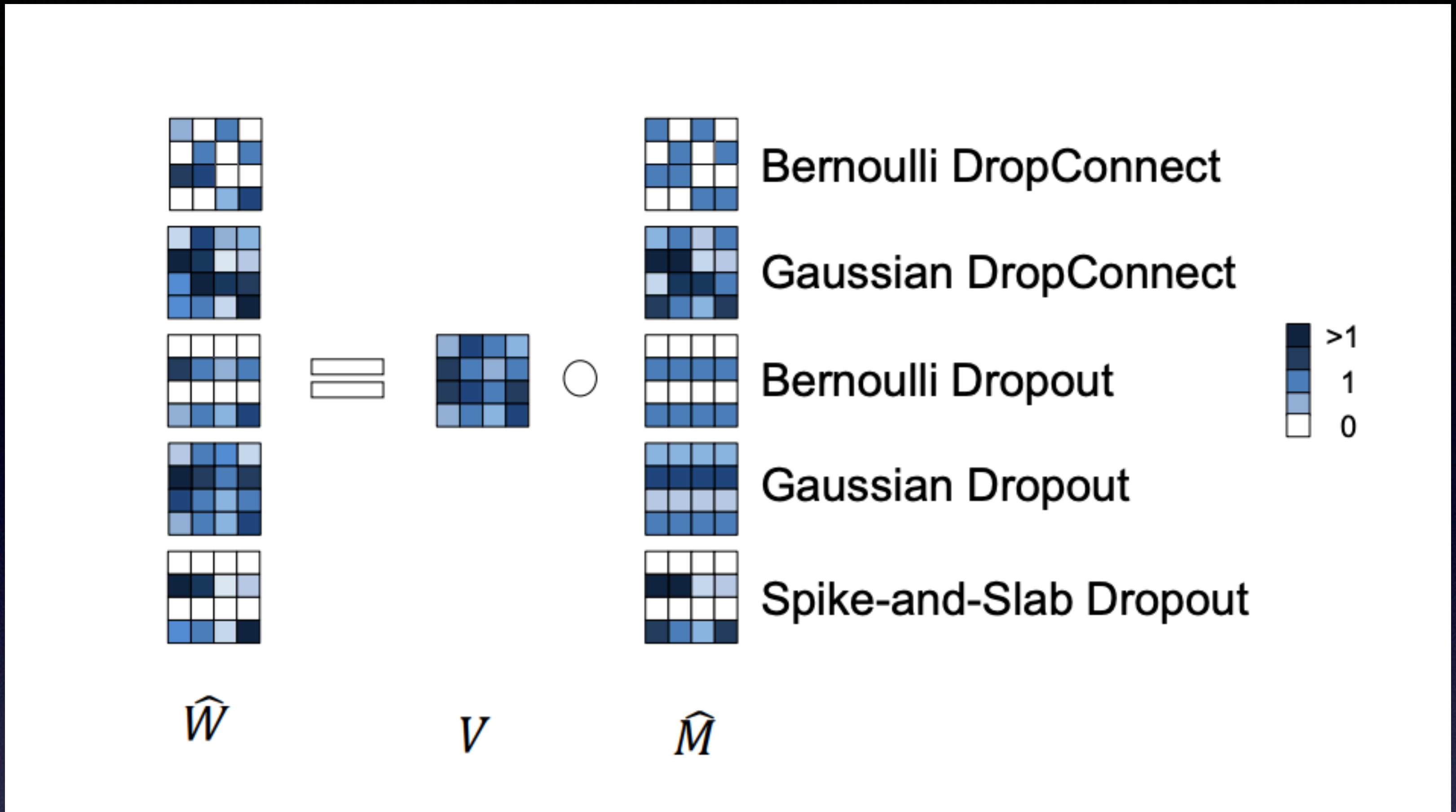
Bernoulli Dropout:
For each line (each neuron), it samples from a Bernoulli distribution.
If the result is 1, it keeps the neuron. If 0 it removes it.

Bernoulli Dropconnect:
For each connection, it samples from a Bernoulli distribution.
If the result is 1, the connections is kept. If 0 it is removed.



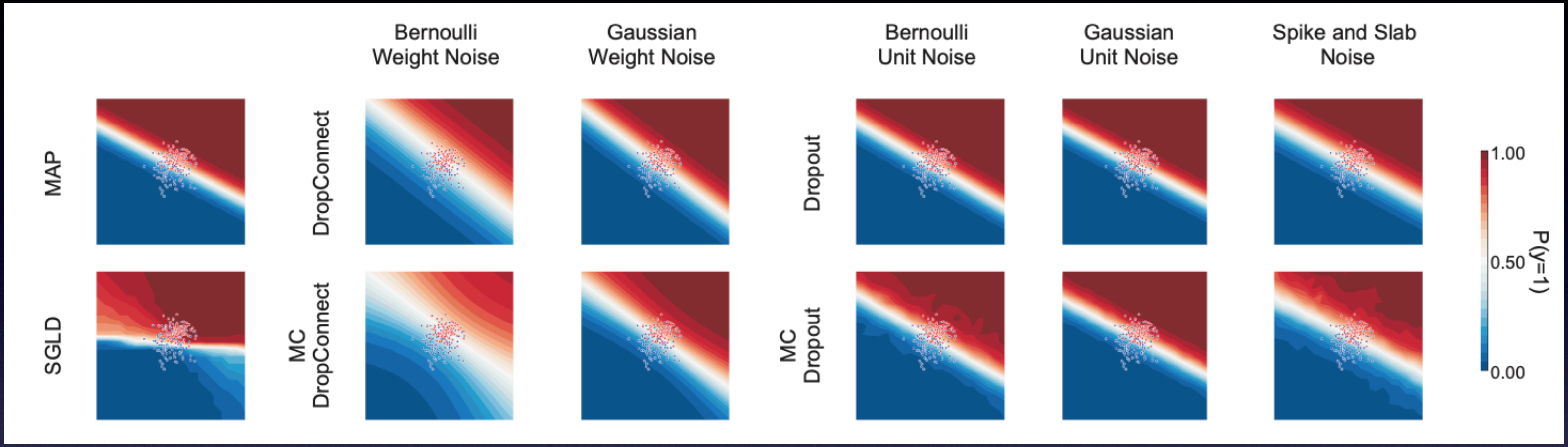
Gaussian Dropout:
For each line (each neuron), it samples from a Gaussian with mean 1.
The value is multiplied to V to sample the weights W.

Gaussian Dropconnect:
For each connection, it samples Gaussian with mean 1.
The value is multiplied to V to sample the weights W.



Spike-and-slab Dropout:
Mixture of Bernoulli Dropout and Gaussian Dropconnect.

Experiments: logistic regression

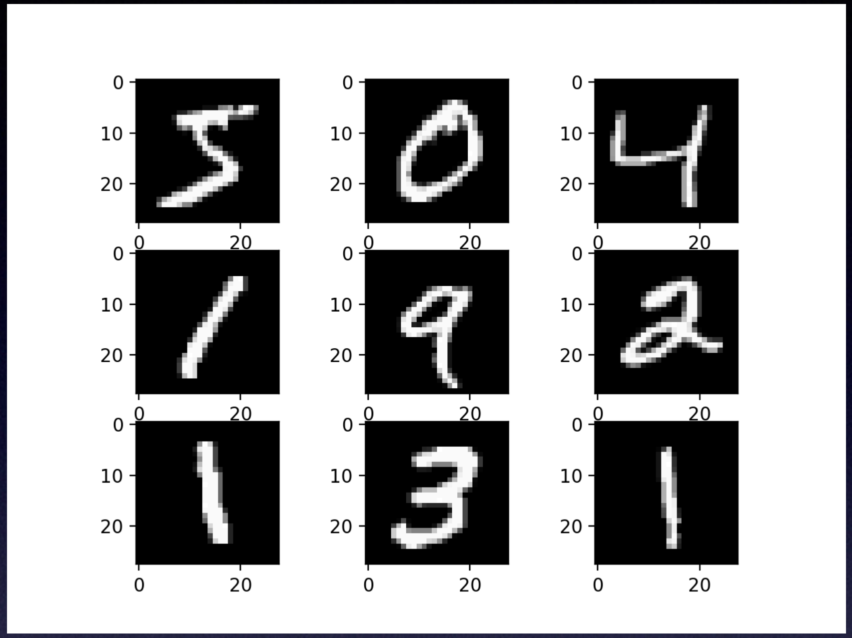


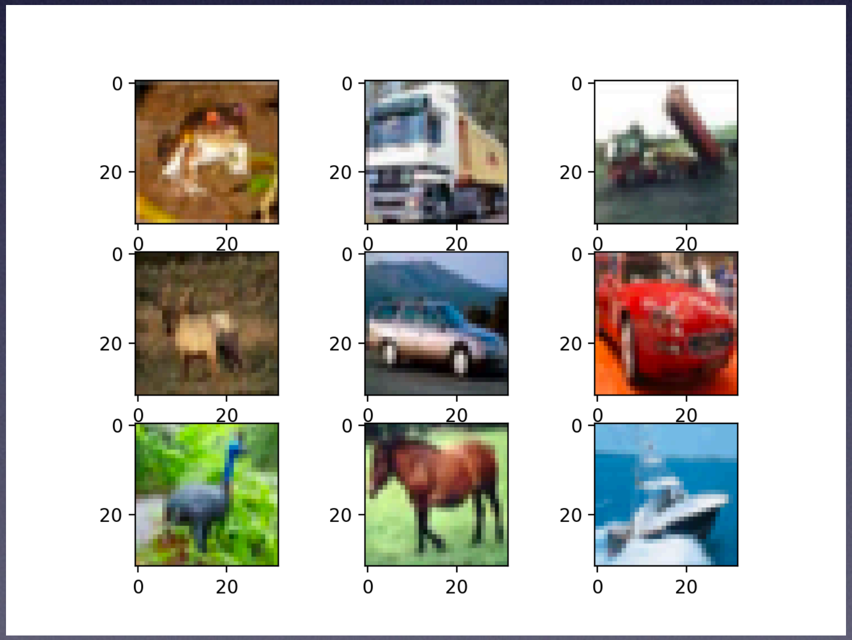
- Linear network with five hidden units
- Classify data drawn from two 2D Gaussian distributions

MAP: Maximum a posteriori. Usual training.
SGLD: Stochastic gradient Langevin Dynamics training.
MC: Monte Carlo, sample multiple models and average predictions.

Experiments: images

2 convolutional layers + FC





13 convolutional layers + FC

Both also use L2 regularisation

Experiments: images

Table 1: MNIST and CIFAR-10 mean and standard deviation of test errors for the trained convolutional neural networks (CNNs) with and without Monte-Carlo (MC) across 5 runs, each MC run using 10 samples.

	MNIST		CIFAR-10	
Method	**Mean Error (%)**	**Error Std. Dev.**	**Mean Error (%)**	**Error Std. Dev.**
MAP	0.76	-	25.86	-
Bernoulli DropConnect	0.56	-	16.46	-
MC Bernoulli DropConnect	0.56	0.03	16.59	0.11
Gaussian DropConnect	0.56	-	16.78	-
MC Gaussian DropConnect	0.58	0.02	16.65	0.11
Bernoulli Dropout	0.49	-	11.23	-
MC Bernoulli Dropout	0.48	0.03	9.95	0.08
Gaussian Dropout	0.42	-	9.07	-
MC Gaussian Dropout	0.36	0.04	9.00	0.10
Spike-and-Slab Dropout	0.48	–	10.64	–
MC Spike-and-Slab Dropout	0.46	0.01	10.05	0.06

Sampling seems to improve little the overall prediction, apart for Bernoulli Dropout.

More interesting test:
add Gaussian noise of increasing variance to test images

Calibration plot

- Classifiers produce class probabilities
- They are typically tested through Precision/Recall/F1

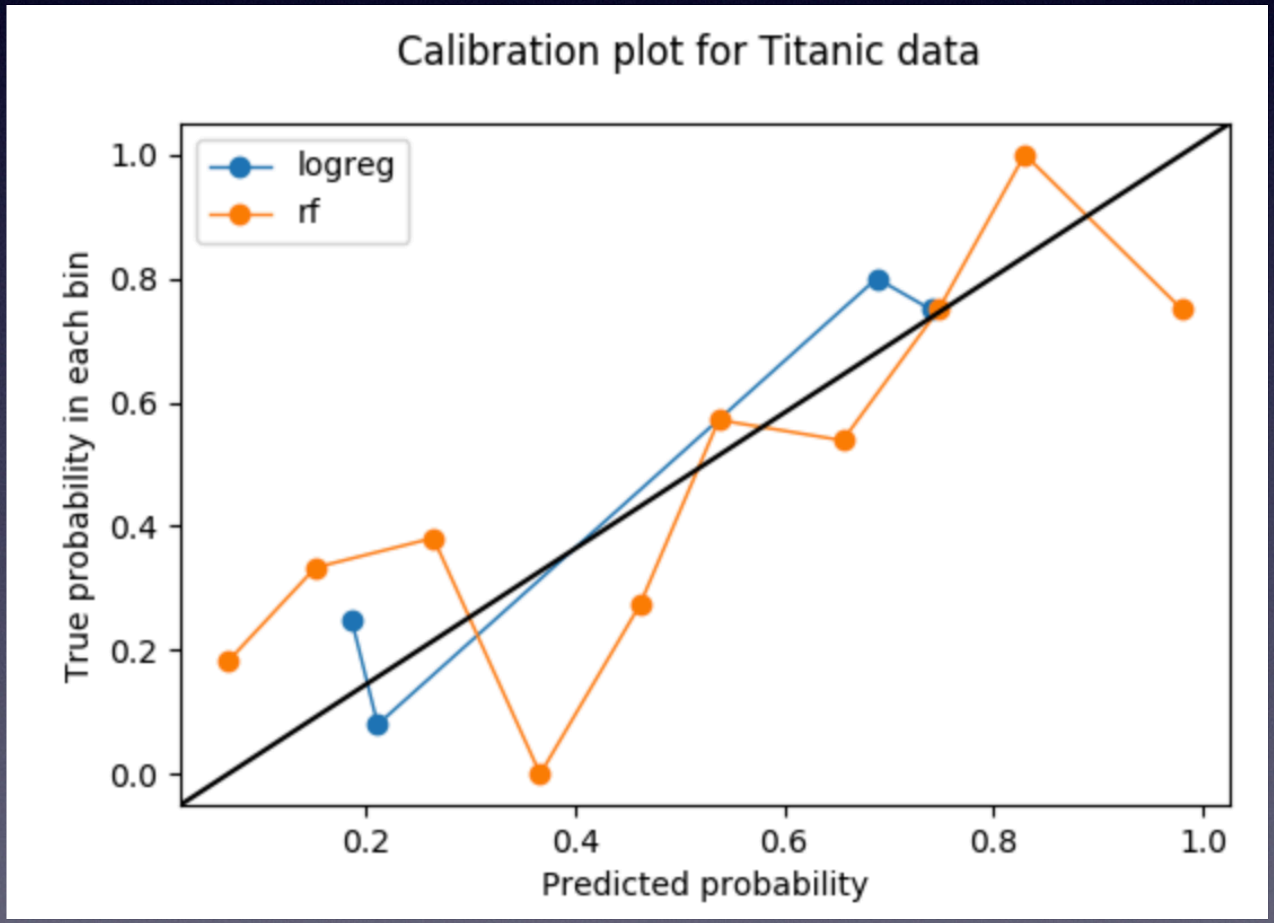
How do I know if I can trust the raw output to be a probability?

Calibration plot

- Classifiers produce class probabilities
- They are typically tested through Precision/Recall/F1

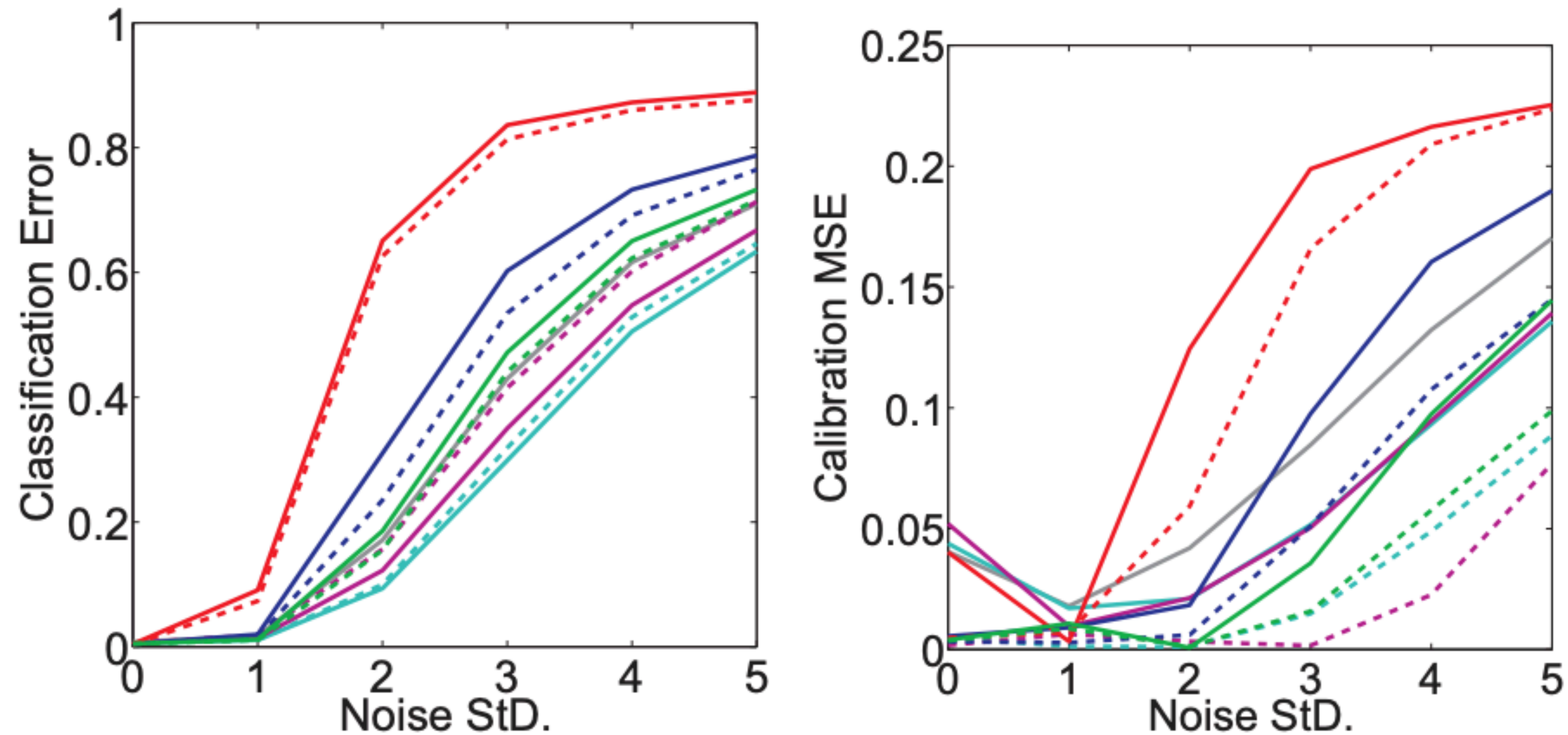
How do I know if I can trust the raw output to be a probability?

Calibration plot:
the y-value is the proportion of true outcomes, and x-value is the mean predicted probability.
Well-calibrated <=> y=x.

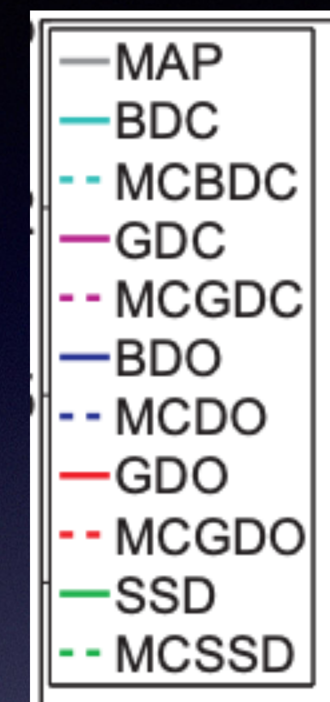


Calibration MSE:
mean squared error between the model prediction and y=x line

Experiments: MNIST

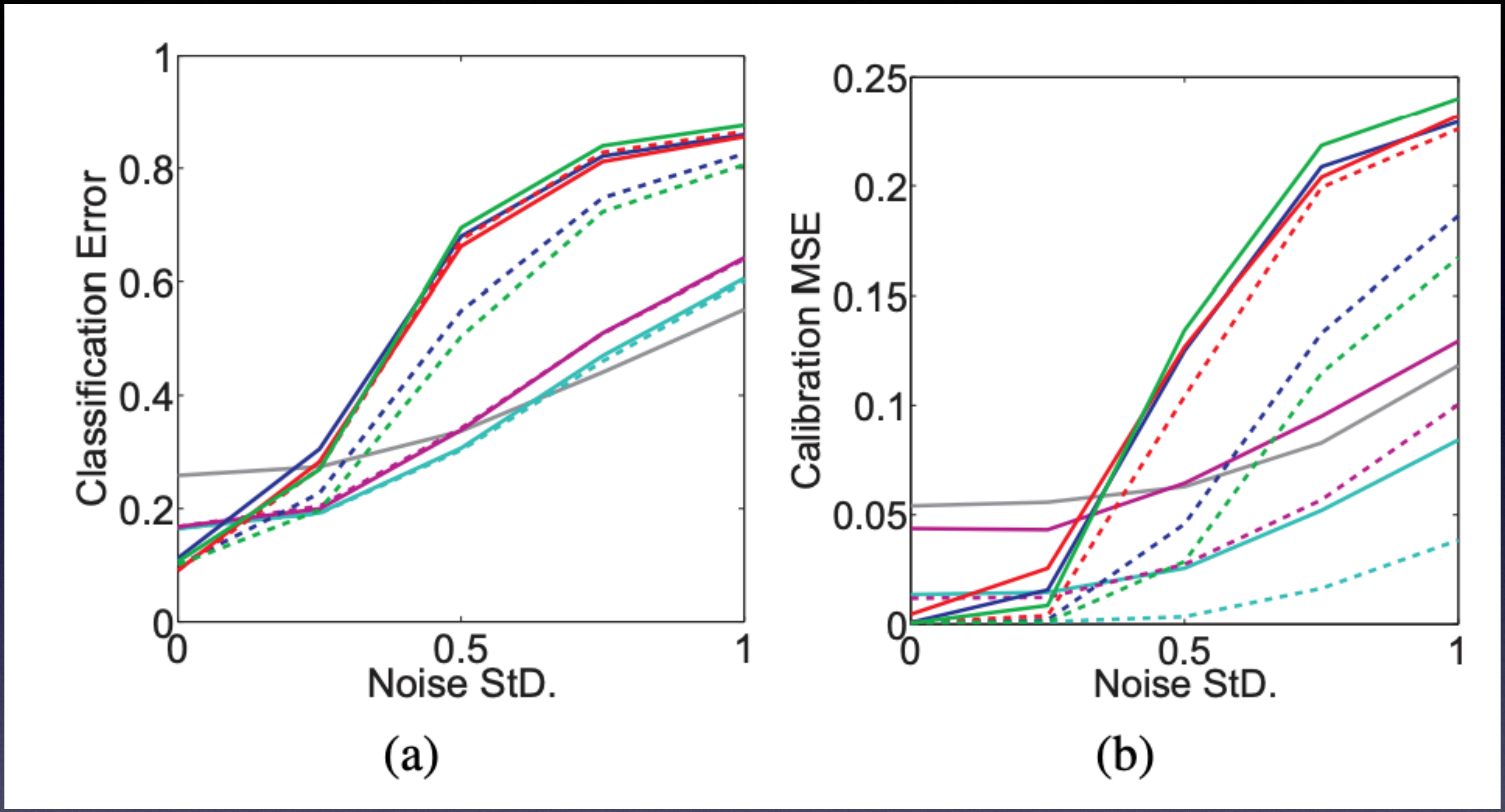


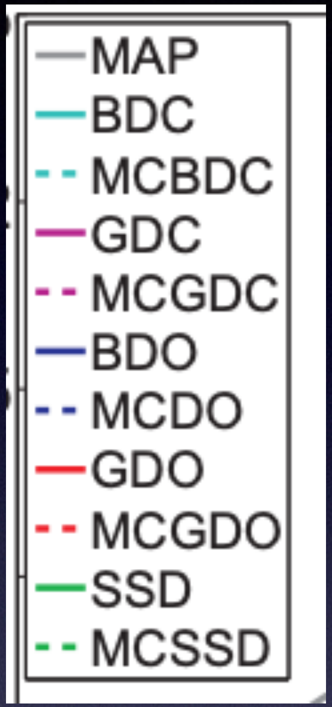
(a) (b)



BDC, BDO: Bernoulli DropConnect and Dropout
GDC, GDO: Gaussian DropConnect and Dropout
SSD: Spike-and-slab Dropout

Experiments: Cifar-10





BDC, BDO: Bernoulli DropConnect and Dropout
GDC, GDO: Gaussian DropConnect and Dropout
SSD: Spike-and-slab Dropout

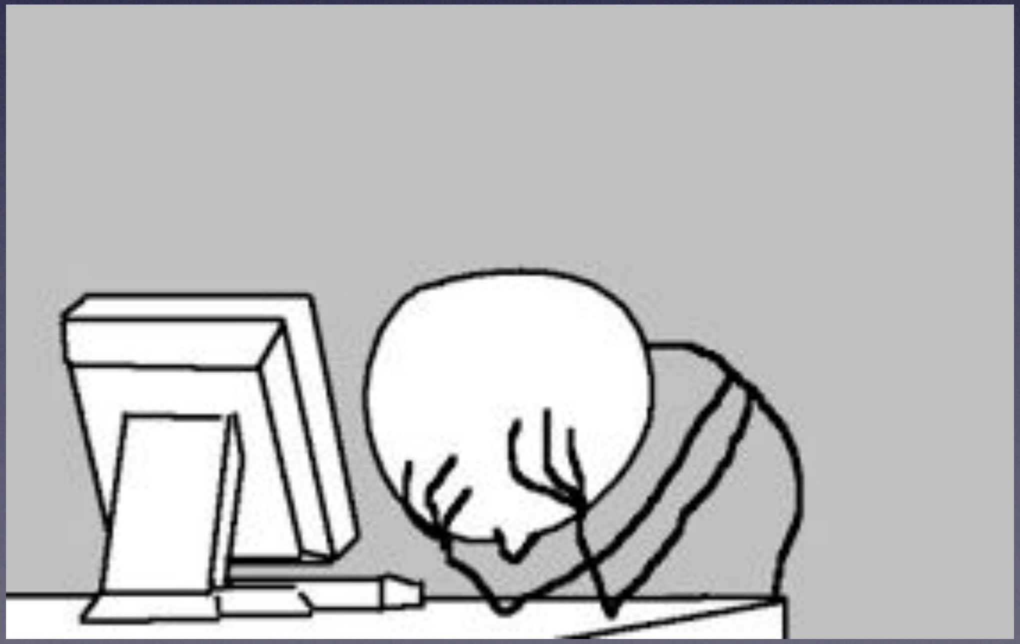
Conclusions on
"Robustly representing uncertainty through sampling in deep neural networks"

- DropConnect seems to yield better calibration than Dropout
- Sampling seems to make models more robust to noise

Conclusions on
"Robustly representing uncertainty through sampling in deep neural networks"

- DropConnect seems to yield better calibration than Dropout
- Sampling seems to make models more robust to noise

However:
- The examples in the paper leave more questions than answers
- We are not directly comparing uncertainty estimation, just calibration.



So… is it any good?!!

More recent example

scientific reports

Check for updates

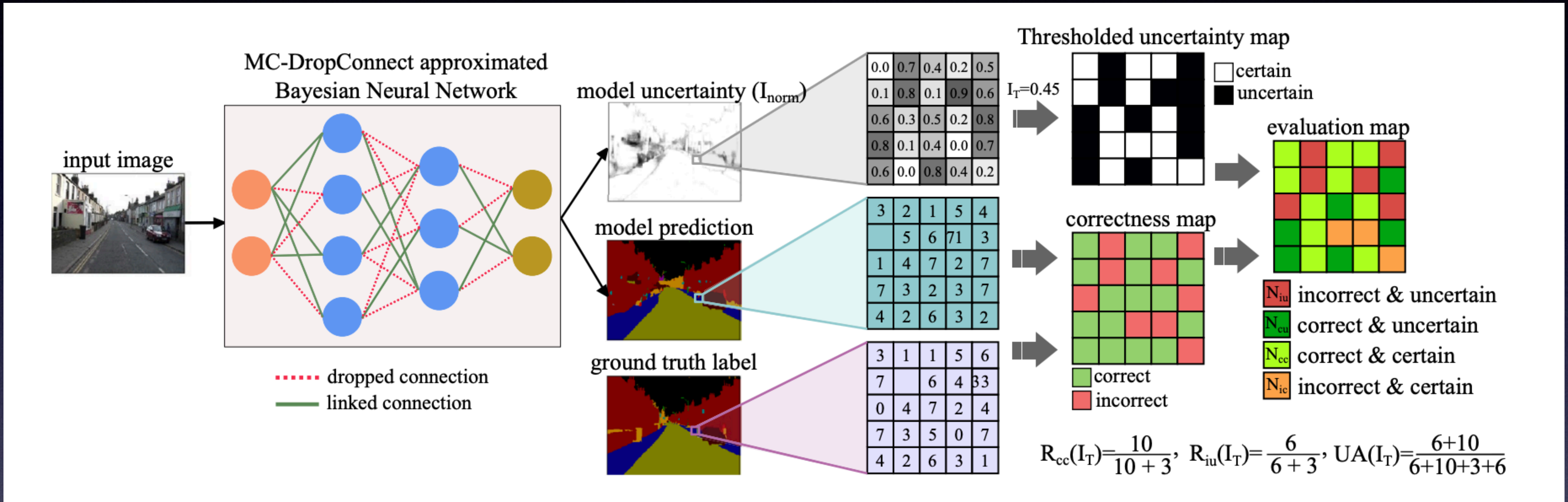
OPEN

DropConnect is effective in modeling uncertainty of Bayesian deep networks

Aryan Mobiny[1]✉, Pengyu Yuan[1], Supratik K. Moulik[2], Naveen Garg[3], Carol C. Wu[3] & Hien Van Nguyen[1]

- Published in 2021
- Applies MC DropConnect to semantic segmentation
- Shows improvement of DropConnect over Dropout

More recent example



Segmentation <=> pixel-wise classification
They test this approach also on MNIST and Cifar-10

Test on MNIST and Cifar-10

They take different test metrics

$$R_{cc}(I_T) = P_{I_T}(\text{correct}|\text{certain}) = \frac{P(\text{correct}, \text{certain})}{P(\text{certain})} = \frac{\mathrm{N}_{cc}}{\mathrm{N}_{cc} + \mathrm{N}_{ic}}$$

$$R_{iu}(I_T) = P_{I_T}(\text{uncertain}|\text{incorrect}) = \frac{P(\text{uncertain}, \text{incorrect})}{P(\text{incorrect})} = \frac{\mathrm{N}_{iu}}{\mathrm{N}_{iu} + \mathrm{N}_{ic}}$$

$$\mathrm{UA}(I_T) = \frac{\mathrm{N}_{cc} + \mathrm{N}_{iu}}{\mathrm{N}_{cc} + \mathrm{N}_{iu} + \mathrm{N}_{cu} + \mathrm{N}_{ic}}$$

Test on MNIST and Cifar-10

They take different test metrics

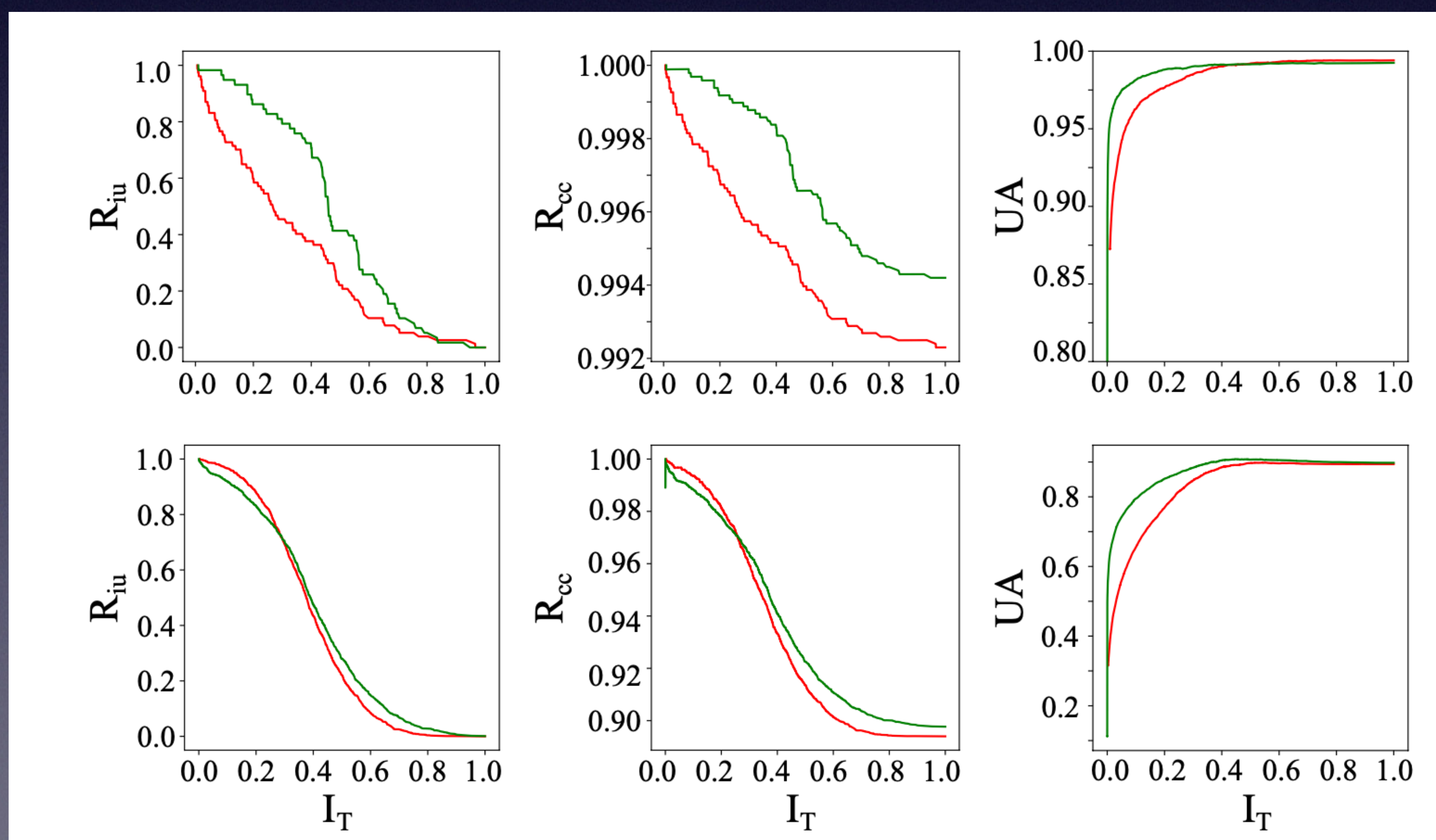
$$R_{cc}(I_T) = P_{I_T}(\text{correct}|\text{certain}) = \frac{P(\text{correct, certain})}{P(\text{certain})} = \frac{\mathrm{N}_{cc}}{\mathrm{N}_{cc} + \mathrm{N}_{ic}}$$

$$R_{iu}(I_T) = P_{I_T}(\text{uncertain}|\text{incorrect}) = \frac{P(\text{uncertain, incorrect})}{P(\text{incorrect})} = \frac{\mathrm{N}_{iu}}{\mathrm{N}_{iu} + \mathrm{N}_{ic}}$$

$$\mathrm{UA}(I_T) = \frac{\mathrm{N}_{cc} + \mathrm{N}_{iu}}{\mathrm{N}_{cc} + \mathrm{N}_{iu} + \mathrm{N}_{cu} + \mathrm{N}_{ic}}$$

MNIST

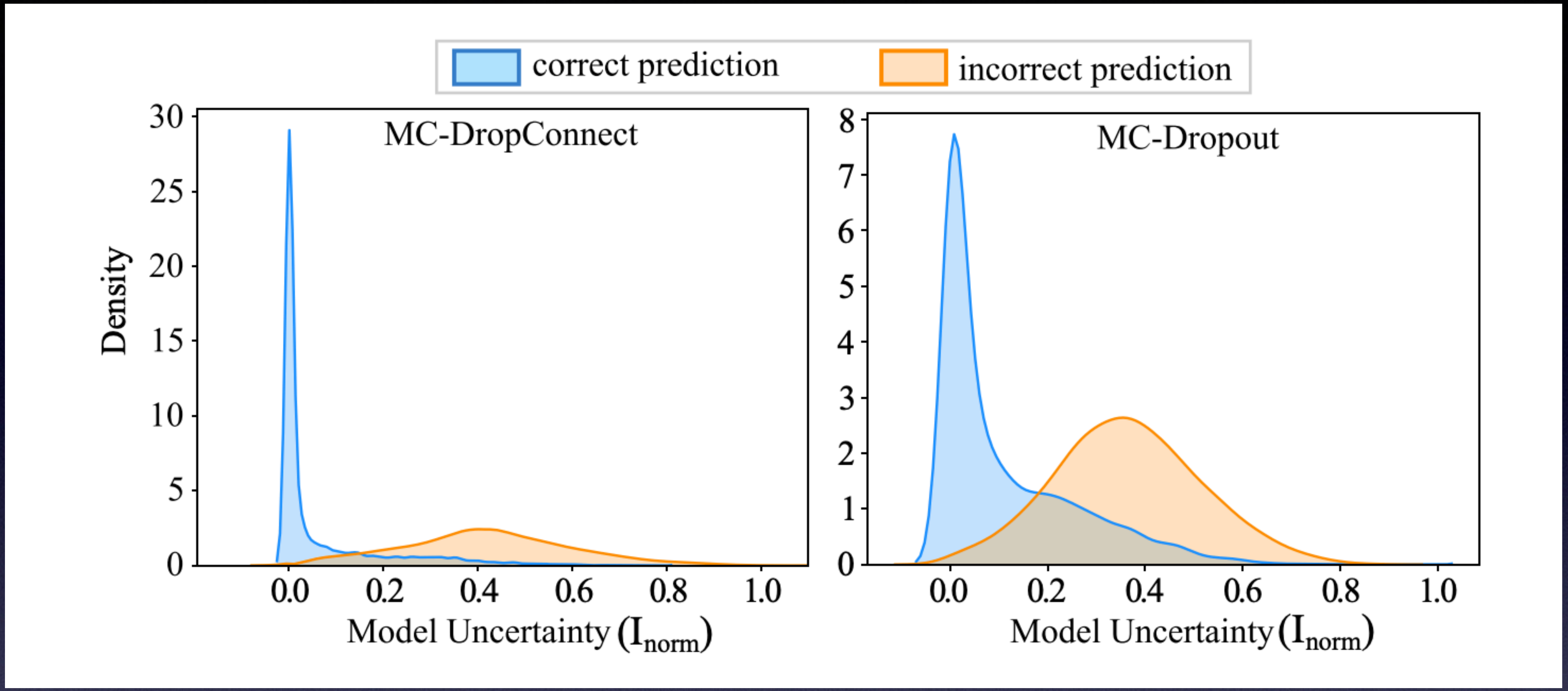
Cifar-10



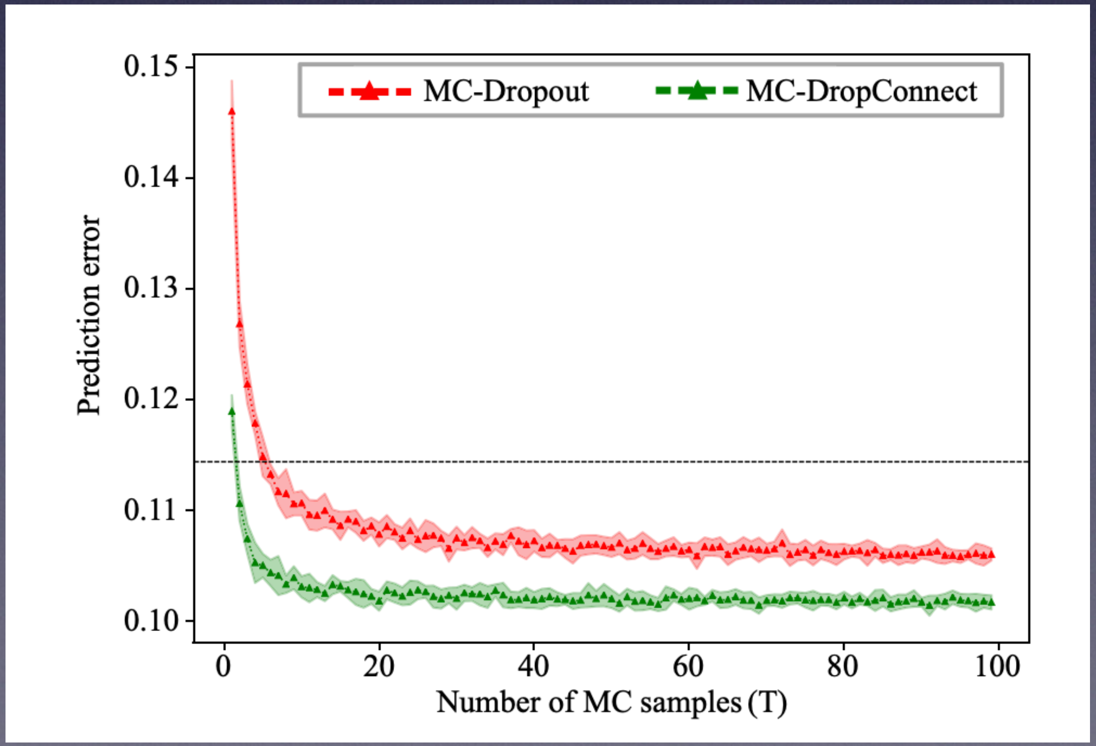
Green: DropConnect

Red: DropOut

Image segmentation



- Incorrect predictions have higher uncertainty
- DropConnect does a better job at uncertainty estimation
- **Code is available!** github.com/hula-ai/mc_dropconnect



Overall Conclusions

- Epistemic uncertainty can be estimated through sampling

- Uncertainty values cannot be directly interpreted as probability, but rather give relative confidence on prediction of one model over another (uncertainty threshold)

- This may be why there are very few papers using it for regression

- Calibration seems to benefit greatly from resampling

- DropConnect seems to beat Dropout in uncertainty estimation

Overall Conclusions

- Epistemic uncertainty can be estimated through sampling

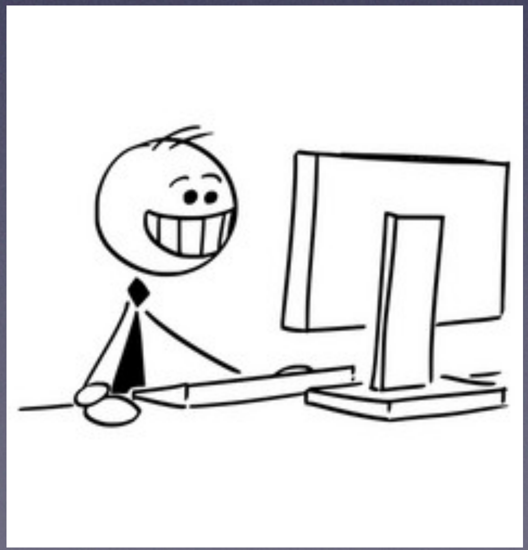
- Uncertainty values cannot be directly interpreted as probability, but rather give relative confidence on prediction of one model over another (uncertainty threshold)

- This may be why there are very few papers using it for regression

- Calibration seems to benefit greatly from resampling

- DropConnect seems to beat Dropout in uncertainty estimation

However:
- Uncertainty through resampling needs bigger model for same accuracy
- Still useful when model size is not too much of a constraint



Questions?