UNIVERSITY OF CAMBRIDGE Yandex Research

Reverse KL-Divergence Training of Prior Networks: Improved Uncertainty and Adversarial Robustness

Andrey Malinin and Mark Gales

18 October 2019

Joint work with...

(a) Prof. Mark Gales

Overview of the Talk

1. Context: Why do we need Uncertainty Estimation?
2. RECAP: Sources of Uncertainty in Predictions
3. RECAP: Ensemble Approaches
4. Prior Networks
5. Adversarial Attack Detection

Overview of the Talk

1. **Context: Why do we need Uncertainty Estimation?**
2. Sources of Uncertainty in Predictions
3. Ensemble Approaches
4. Prior Networks
5. Adversarial Attack Detection

Scenario

- Given a deployed model and a test input $\boldsymbol{x}^*$ we wish to:
  - Obtain a prediction
  - Obtain a measure of **uncertainty in prediction**
- Take action based estimate of uncertainty
  - Reject prediction / stop decoding sentence
  - Ask for human intervention
  - Use active learning

Applications of Uncertainty Estimation

- Uncertainty should be assessed in the context of an **application**
- Threshold-based outlier detection $\rightarrow$
  - Misclassification Detection [Hendrycks and Gimpel, 2016]
  - Out-of-distribution input Detection [Malinin and Gales, 2019]
  - Adversarial Attack Detection [Malinin and Gales, 2019]
- Active Learning [Gal, 2016]
- Reinforcement Learning uncertainty-driven exploration [Osband et al., 2016]
- Other...

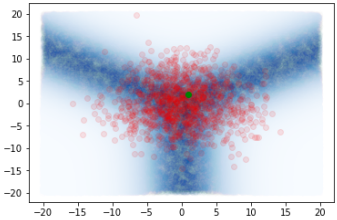
Assessment of Uncertainty Quality

- Uncertainty should be assessed in the context of an **application**
- Threshold-based outlier detection $\rightarrow$
  - Misclassification Detection [Hendrycks and Gimpel, 2016]
  - **Out-of-distribution input Detection** [Malinin and Gales, 2019]
  - **Adversarial Attack Detection** [Malinin and Gales, 2019]
- Active Learning [Gal, 2016]
- Reinforcement Learning uncertainty-driven exploration [Osband et al., 2016]
- Other...

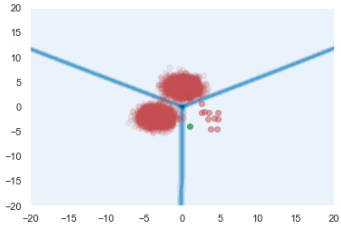
Overview of the Talk

1. Context: Why do we need Uncertainty Estimation?
2. **Sources of Uncertainty in Predictions**
3. Ensemble Approaches
4. Prior Networks
5. Adversarial Attack Detection

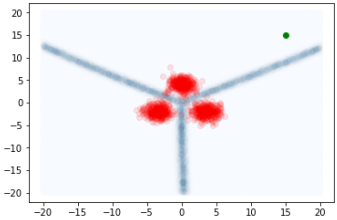
Sources of Uncertainty



(a) Data Uncertainty



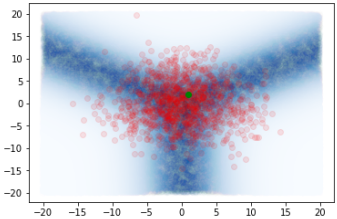
(b) Data Sparsity



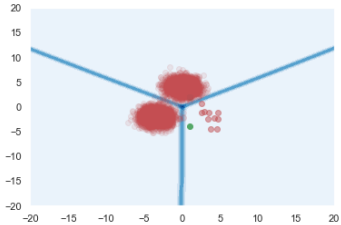
(c) Out-of-Distribution inputs

- Knowledge (epistemic) uncertainty refers to both:
  - Data Sparsity and Out-of-distribution inputs

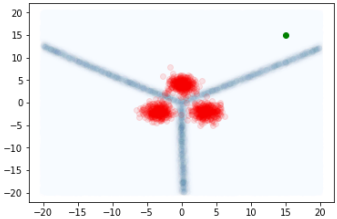
Sources of Uncertainty



(a) Data Uncertainty



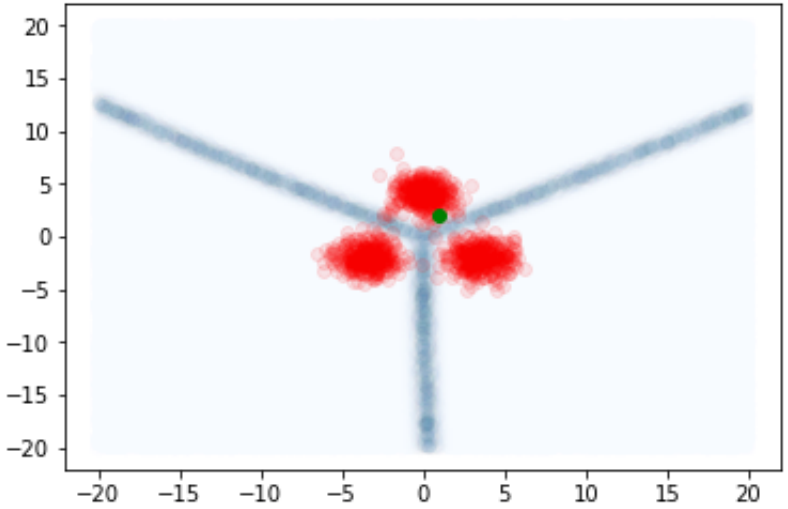
(b) Knowledge Uncertainty



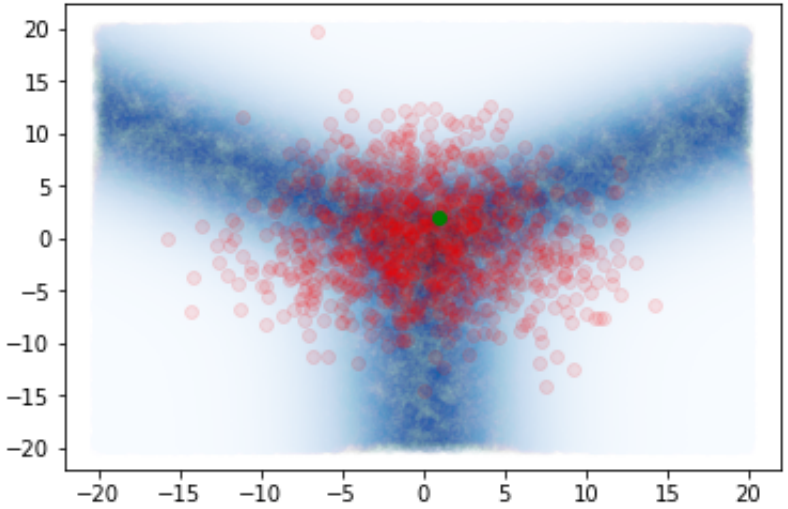
(c) Knowledge Uncertainty

- Knowledge (epistemic) uncertainty refers to both:
  - Data Sparsity and Knowledge Uncertainty

Data (Aleatoric) Uncertainty

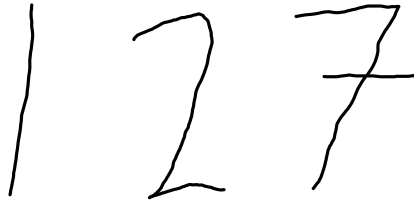


Data Uncertainty



Data Uncertainty

- Distinct Classes



- Overlapping Classes

Data Uncertainty

- Data Uncertainty $\rightarrow$ **Known-Unknown**
- Uncertainty due to properties of data
  - Class overlap (complexity of decision boundaries)
  - Human labelling error

Data Uncertainty

- Data Uncertainty is the *entropy* of the *true data distribution* $\rightarrow$

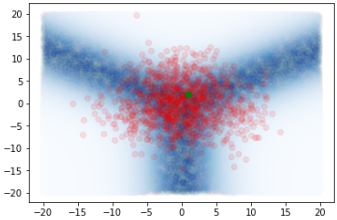
$$\mathcal{H}[\mathrm{P}_{\mathtt{tr}}(y|\boldsymbol{x}^*)] = -\sum_{c=1}^{K} \mathrm{P}_{\mathtt{tr}}(y=\omega_c|\boldsymbol{x}^*) \ln \mathrm{P}_{\mathtt{tr}}(y=\omega_c|\boldsymbol{x}^*)$$

- Captured by the entropy of a model's posterior over classes $\rightarrow$

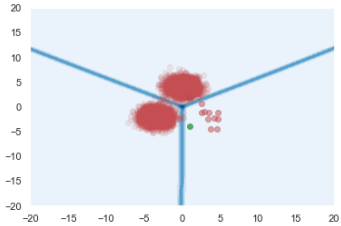
$$\mathcal{H}[\mathrm{P}(y|\boldsymbol{x}^*, \hat{\boldsymbol{\theta}})] = -\sum_{c=1}^{K} \mathrm{P}(y=\omega_c|\boldsymbol{x}^*, \hat{\boldsymbol{\theta}}) \ln \mathrm{P}(y=\omega_c|\boldsymbol{x}^*, \hat{\boldsymbol{\theta}})$$

- Data Uncertainty is captured as a consequence of Maximum Likelihood Estimation

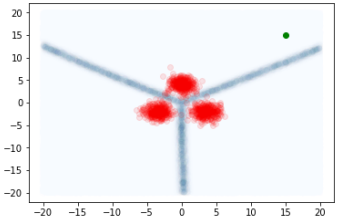
Sources of Uncertainty



(a) Data Uncertainty



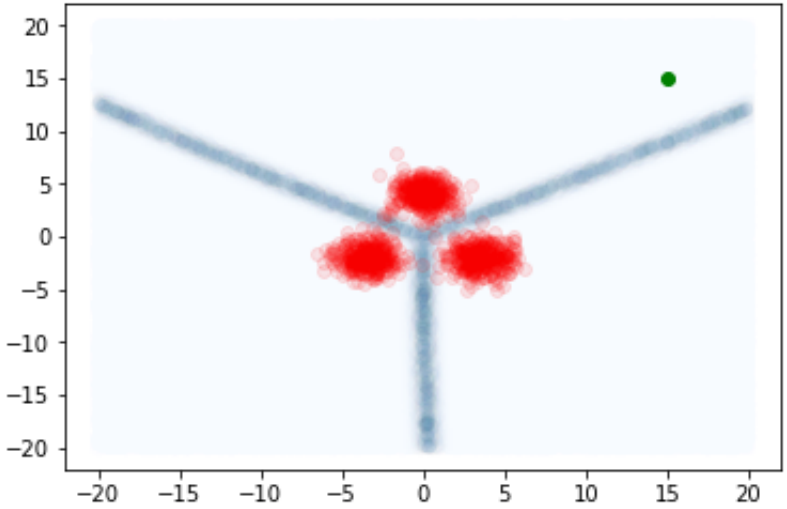
(b) Data Sparsity



(c) Out-of-Distribution inputs

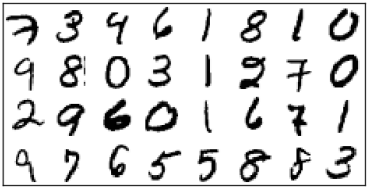
- Knowledge (epistemic) uncertainty refers to both:
  - Data Sparsity and Out-of-distribution inputs

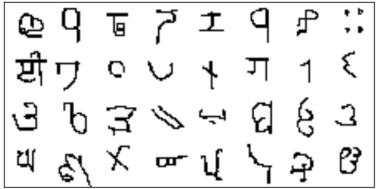
Knowledge Uncertainty



Knowledge Uncertainty - Out-of-Distribution

- Unseen classes





- Unseen variations of seen classes

Sources of Uncertainty

- Data Uncertainty $\rightarrow$ **Known-Unknown**
  - Class overlap (complexity of decision boundaries)
  - Human labelling error
- Knowledge Uncertainty $\rightarrow$ **Unknown-Unknown**
  - Test input in out-of-distribution region far from training data
- Appropriate **action** depends on **source** of uncertainty
  - Separating sources of uncertainty requires **Ensemble approaches**
  - ... or Prior Networks

Overview of the Talk

1. Context: Why do we need Uncertainty Estimation?
2. Sources of Uncertainty in Predictions
3. **Ensemble Approaches**
4. Prior Networks
5. Assessment of Uncertainty Quality

Ensemble Approaches

- Uncertainty in $\boldsymbol{\theta}$ captured by model posterior $\mathrm{p}(\boldsymbol{\theta}|\mathcal{D}) \rightarrow$

$$\mathrm{p}(\boldsymbol{\theta}|\mathcal{D}) = \frac{\mathrm{p}(\mathcal{D}|\boldsymbol{\theta})\mathrm{p}(\boldsymbol{\theta})}{\mathrm{p}(\mathcal{D})}$$

- Bayesian inference of $\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta}) \rightarrow$

$$\mathrm{P}(y|\boldsymbol{x}^*, \mathcal{D}) = \mathbb{E}_{\mathrm{p}(\boldsymbol{\theta}|\mathcal{D})}\left[\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta})\right]$$

- Can consider an ensemble of models $\rightarrow$

$$\{\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta}^{(m)})\}_{m=1}^{M}, \;\; \boldsymbol{\theta}^{(m)} \sim \mathrm{p}(\boldsymbol{\theta}|\mathcal{D})$$

- Choose desired behaviour of ensemble via prior $\mathrm{p}(\boldsymbol{\theta})$

Total Uncertainty

- Consider the entropy of the predictive posterior $\mathrm{P}(y|\boldsymbol{x}^*, \mathcal{D}) \rightarrow$

$$\begin{aligned}\mathcal{H}\big[\mathrm{P}(y|\boldsymbol{x}^*, \mathcal{D})\big] &= \mathcal{H}\big[\mathbb{E}_{p(\boldsymbol{\theta}|\mathcal{D})}[\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta})]\big] \\ &\approx \mathcal{H}\Big[\frac{1}{M}\sum_{m=1}^{M}\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta}^{(m)})\Big],\ \boldsymbol{\theta}^{(m)} \sim \mathrm{p}(\boldsymbol{\theta}|\mathcal{D})\end{aligned}$$

- Measure of Total Uncertainty
  - Combination of Data uncertainty and **Knowledge uncertainty**

Expected Data Uncertainty

- Lets consider an ensemble of models $\{\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta}^{(m)})\}_{m=1}^{M}, \ \boldsymbol{\theta}^{(m)} \sim \mathrm{p}(\boldsymbol{\theta}|\mathcal{D})$
  - Each model $\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta}^{(m)})$ captures an different estimate of data uncertainty.
- Ensemble estimate of data uncertainty $\rightarrow$ Expected Data Uncertainty

$$\mathbb{E}_{p(\boldsymbol{\theta}|\mathcal{D})}\big[\mathcal{H}[\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta})]\big] \approx \frac{1}{M}\sum_{m=1}^{M} \mathcal{H}\big[\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta}^{(m)})\big], \ \boldsymbol{\theta}^{(m)} \sim \mathrm{p}(\boldsymbol{\theta}|\mathcal{D})$$

- Not the same as entropy of the predictive posterior $\mathrm{P}(y|\boldsymbol{x}^*, \mathcal{D})$

Model Uncertainty

- If the predictions from the models are consistent

$$\underbrace{\mathcal{H}\left[\mathbb{E}_{\mathrm{p}(\boldsymbol{\theta}|\mathcal{D})}[\mathrm{P}(y|\boldsymbol{x}^*,\boldsymbol{\theta})]\right]}_{\textit{Total Uncertainty}} - \underbrace{\mathbb{E}_{\mathrm{p}(\boldsymbol{\theta}|\mathcal{D})}\left[\mathcal{H}[\mathrm{P}(y|\boldsymbol{x}^*,\boldsymbol{\theta})]\right]}_{\textit{Expected Data Uncertainty}} = 0$$

- If the predictions from the models are diverse

$$\underbrace{\mathcal{H}\left[\mathbb{E}_{\mathrm{p}(\boldsymbol{\theta}|\mathcal{D})}[\mathrm{P}(y|\boldsymbol{x}^*,\boldsymbol{\theta})]\right]}_{\textit{Total Uncertainty}} - \underbrace{\mathbb{E}_{\mathrm{p}(\boldsymbol{\theta}|\mathcal{D})}\left[\mathcal{H}[\mathrm{P}(y|\boldsymbol{x}^*,\boldsymbol{\theta})]\right]}_{\textit{Expected Data Uncertainty}} > 0$$

- Difference of the two is a measure of model uncertainty

$$\underbrace{\mathcal{I}[y,\boldsymbol{\theta}|\boldsymbol{x}^*,\mathcal{D}]}_{\textit{Model Uncertainty}} = \underbrace{\mathcal{H}\left[\mathbb{E}_{\mathrm{p}(\boldsymbol{\theta}|\mathcal{D})}[\mathrm{P}(y|\boldsymbol{x}^*,\boldsymbol{\theta})]\right]}_{\textit{Total Uncertainty}} - \underbrace{\mathbb{E}_{\mathrm{p}(\boldsymbol{\theta}|\mathcal{D})}\left[\mathcal{H}[\mathrm{P}(y|\boldsymbol{x}^*,\boldsymbol{\theta})]\right]}_{\textit{Expected Data Uncertainty}}$$

Model Uncertainty $\rightarrow$ Knowledge Uncertainty

- If the predictions from the models are consistent

$$\underbrace{\mathcal{H}\left[\mathbb{E}_{\mathrm{p}(\boldsymbol{\theta}|\mathcal{D})}[\mathrm{P}(y|\boldsymbol{x}^*,\boldsymbol{\theta})]\right]}_{\textit{Total Uncertainty}} - \underbrace{\mathbb{E}_{\mathrm{p}(\boldsymbol{\theta}|\mathcal{D})}\left[\mathcal{H}[\mathrm{P}(y|\boldsymbol{x}^*,\boldsymbol{\theta})]\right]}_{\textit{Expected Data Uncertainty}} = 0$$

- If the predictions from the models are diverse

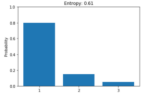
$$\underbrace{\mathcal{H}\left[\mathbb{E}_{\mathrm{p}(\boldsymbol{\theta}|\mathcal{D})}[\mathrm{P}(y|\boldsymbol{x}^*,\boldsymbol{\theta})]\right]}_{\textit{Total Uncertainty}} - \underbrace{\mathbb{E}_{\mathrm{p}(\boldsymbol{\theta}|\mathcal{D})}\left[\mathcal{H}[\mathrm{P}(y|\boldsymbol{x}^*,\boldsymbol{\theta})]\right]}_{\textit{Expected Data Uncertainty}} > 0$$

- Difference of the two is a measure of knowledge uncertainty

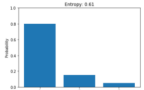
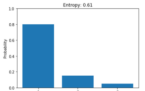
$$\underbrace{\mathcal{I}[y,\boldsymbol{\theta}|\boldsymbol{x}^*,\mathcal{D}]}_{\textit{Knowledge Uncertainty}} = \underbrace{\mathcal{H}\left[\mathbb{E}_{\mathrm{p}(\boldsymbol{\theta}|\mathcal{D})}[\mathrm{P}(y|\boldsymbol{x}^*,\boldsymbol{\theta})]\right]}_{\textit{Total Uncertainty}} - \underbrace{\mathbb{E}_{\mathrm{p}(\boldsymbol{\theta}|\mathcal{D})}\left[\mathcal{H}[\mathrm{P}(y|\boldsymbol{x}^*,\boldsymbol{\theta})]\right]}_{\textit{Expected Data Uncertainty}}$$

Ensemble for certain in-domain input

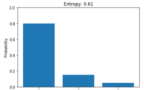
$$\mathrm{p}(\boldsymbol{\theta}) + \mathcal{D} \to \mathrm{p}(\boldsymbol{\theta}|\mathcal{D}) \longrightarrow$$

$\boldsymbol{\theta}^{(1)} \longrightarrow$  $\rightarrow$

$\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta}^{(1)})$

$\boldsymbol{\theta}^{(i)} \longrightarrow$  $\rightarrow$ 

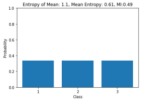
$\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta}^{(i)})$ $\quad$ $\mathrm{P}(y|\boldsymbol{x}^*, \mathcal{D})$

$\boldsymbol{\theta}^{(N)} \longrightarrow$  $\rightarrow$

$\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta}^{(N)})$

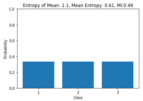
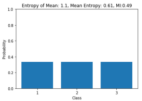
Ensemble for uncertain in-domain input

$$\mathrm{p}(\boldsymbol{\theta}) + \mathcal{D} \rightarrow \mathrm{p}(\boldsymbol{\theta}|\mathcal{D}) \longrightarrow$$

$\boldsymbol{\theta}^{(1)} \longrightarrow$  $\rightarrow$

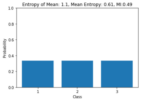
$\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta}^{(1)})$

$\vdots$ $\vdots$

$\boldsymbol{\theta}^{(i)} \longrightarrow$  $\rightarrow$ 

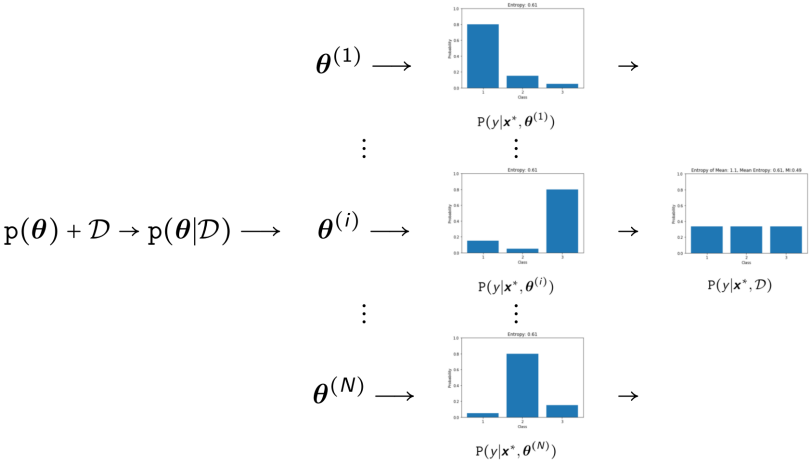
$\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta}^{(i)})$ $\mathrm{P}(y|\boldsymbol{x}^*, \mathcal{D})$

$\vdots$ $\vdots$

$\boldsymbol{\theta}^{(N)} \longrightarrow$  $\rightarrow$

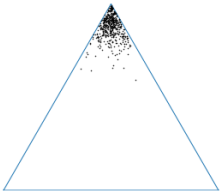
$\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta}^{(N)})$

Ensemble for Out-of-Domain input

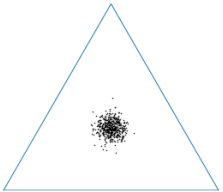


Distributions on a Simplex

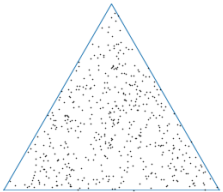
- Ensemble $\{\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta}^{(m)})\}_{m=1}^{M}$ can be visualized on a simplex for an input $\boldsymbol{x}^*$



(a) Confident



(b) Data Uncertainty



(c) Knowledge Uncertainty

Approximate Inference

- Ideally - compute all measures of uncertainty in closed form...
  - But inference is **intractable** for neural networks
  - Bayes' Rule is **intractable** for neural networks
- Solutions $\rightarrow$ use approximate inference
  - Compute approximate posterior $\mathrm{q}(\boldsymbol{\theta}) \approx \mathrm{p}(\boldsymbol{\theta}|\mathcal{D})$
  - Use variational approximations to measures of uncertainty
  - Use Monte-Carlo approximations to measures of uncertainty

Approximate Inference

- Variational Inference:
  - Bayes by Backprop [Blundell et al., 2015]
  - Probabalistic Backpropagation [Hernández-Lobato and Adams, 2015]
- Monte-Carlo Methods:
  - Monte-Carlo Dropout [Gal, 2016, Gal and Ghahramani, 2016]
  - Stochastic Gradient Langevin Dynamics [Welling and Teh, 2011]
  - Fast-Ensembling via Mode Connectivity [Garipov et al., 2018]
  - Stochastic Weight Averaging Gaussian (SWAG) [Maddox et al., 2019]
- Non-Bayesian Ensembles:
  - Bootstrap DQN [Osband et al., 2016]
  - Deep Ensembles [Lakshminarayanan et al., 2017]

Limitations

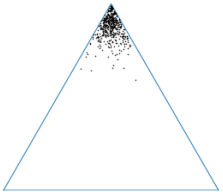
- Hard to guarantee diverse $\{\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta}^{(m)})\}_{m=1}^{M}$ for OOD $\boldsymbol{x}^*$
- Diversity of ensemble depends on:
  - Selection of prior
  - Nature of approximations
  - Architecture of network
  - Properties and size of data
- May be computationally expensive

Overview of the Talk

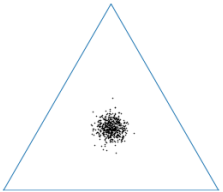
1. Context: Why do we need Uncertainty Estimation?
2. Sources of Uncertainty in Predictions
3. Ensemble Approaches
4. **Prior Networks**
5. Adversarial Attack Detection

Distributions on a Simplex

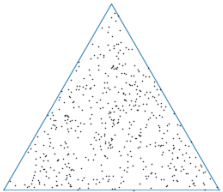
- Ensemble $\{\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta}^{(m)})\}_{m=1}^{M}$ can be visualized on a simplex



(a) Confident



(b) Data Uncertainty



(c) Knowledge Uncertainty

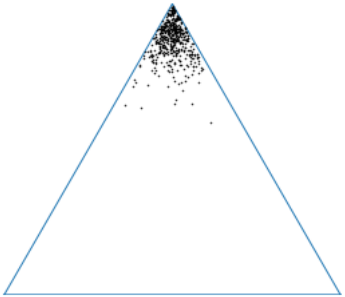
- Same as sampling from **implicit** Distribution over output Distributions

$$\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta}^{(m)}) \sim \mathrm{p}(\boldsymbol{\theta}|\mathcal{D}) \equiv \boldsymbol{\mu}^{(m)} \sim \mathrm{p}(\boldsymbol{\mu}|\boldsymbol{x}^*, \mathcal{D})$$

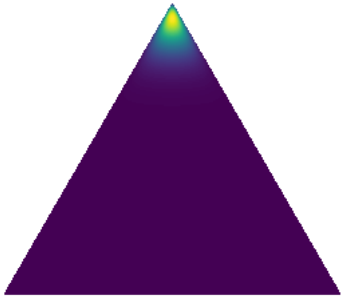
Distributions on a Simplex (cont)

- Expanding out $\boldsymbol{\mu}^{(m)} = \begin{bmatrix} \mathrm{P}(y = \omega_1) \\ \mathrm{P}(y = \omega_2) \\ \vdots \\ \mathrm{P}(y = \omega_K) \end{bmatrix}$, where each $\boldsymbol{\mu}^{(m)}$ is a point on a simplex.

Distribution over Distributions

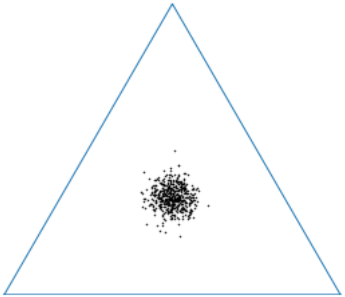


(a) $\{\boldsymbol{\mu}^{(m)}\}_{m=1}^{M}$

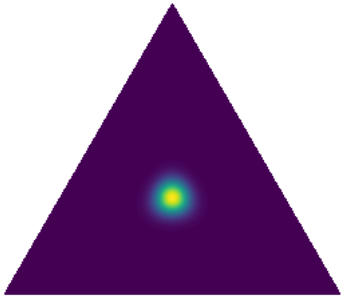


(b) $\mathrm{p}(\boldsymbol{\mu}|\boldsymbol{x}^*, \mathcal{D})$

Distribution over Distributions

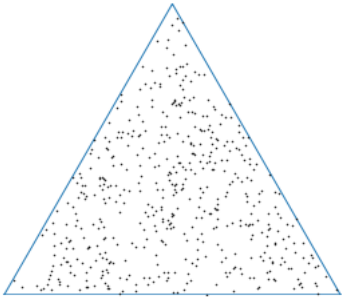


(a) $\{\boldsymbol{\mu}^{(m)}\}_{m=1}^{M}$

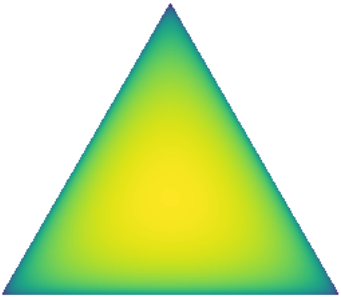


(b) $\mathrm{p}(\boldsymbol{\mu}|\boldsymbol{x}^*, \mathcal{D})$

Distribution over Distributions



(a) $\{\boldsymbol{\mu}^{(m)}\}_{m=1}^{M}$



(b) $\mathrm{p}(\boldsymbol{\mu}|\boldsymbol{x}^*, \mathcal{D})$

Prior Networks [Malinin and Gales, 2018]

- **Explicitly** model $\mathrm{p}(\boldsymbol{\mu}|\boldsymbol{x}^*, \mathcal{D})$ using a Prior Network $\mathrm{p}(\boldsymbol{\mu}|\boldsymbol{x}^*; \hat{\boldsymbol{\theta}})$

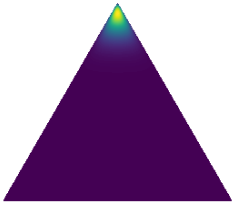
$$\mathrm{p}(\boldsymbol{\mu}|\boldsymbol{x}^*; \hat{\boldsymbol{\theta}}) \approx \mathrm{p}(\boldsymbol{\mu}|\boldsymbol{x}^*, \mathcal{D})$$

- Predictive posterior distribution is given by expected categorical

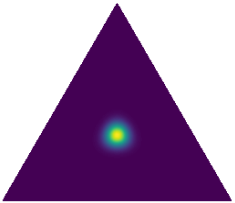
$$\mathrm{P}(y|\boldsymbol{x}^*; \hat{\boldsymbol{\theta}}) = \mathbb{E}_{\mathrm{p}(\boldsymbol{\mu}|\boldsymbol{x}^*; \hat{\boldsymbol{\theta}})}\left[\mathrm{p}(y|\boldsymbol{\mu})\right] = \hat{\boldsymbol{\mu}}$$

Prior Networks [Malinin and Gales, 2018]

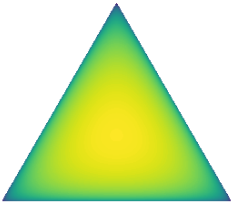
- Construct $\mathrm{p}(\boldsymbol{\mu}|\boldsymbol{x}^*, \hat{\boldsymbol{\theta}})$ to emulate ensemble



(a) Certain

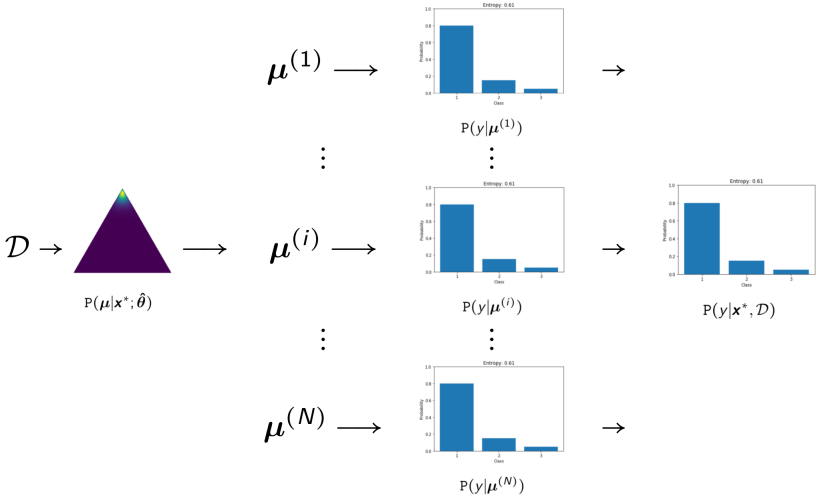


(b) Data Uncertainty

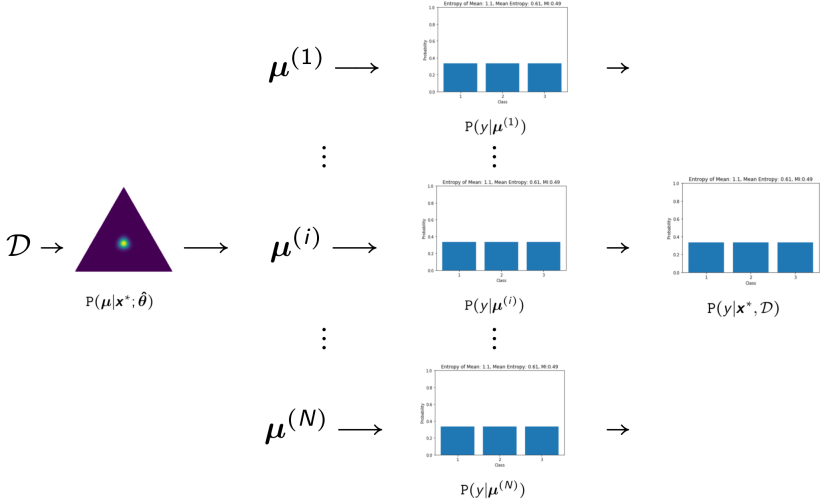


(c) Knowledge Uncertainty

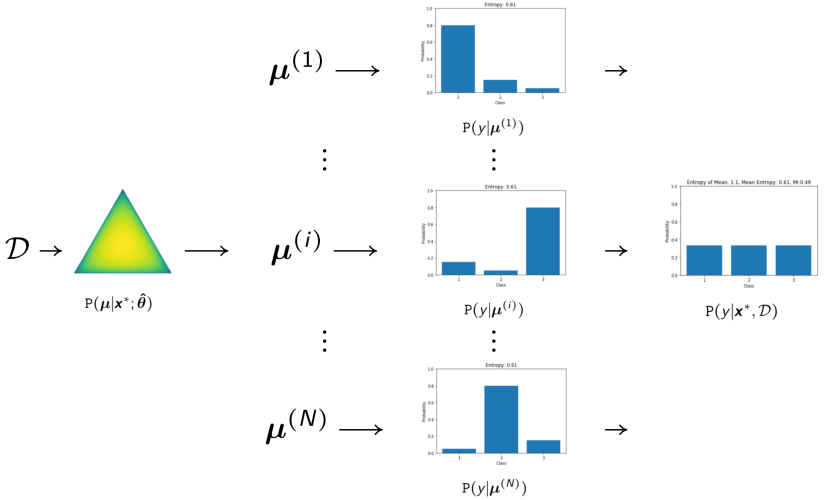
Distributions over Distributions via Prior Networks



Distributions over Distributions via Prior Networks



Distributions over Distributions via Prior Networks



Uncertainty Measures for Prior Networks [Malinin and Gales, 2018]

- Ensemble uncertainty decomposition:

$$\underbrace{\mathcal{I}[y, \boldsymbol{\theta} | \boldsymbol{x}^*, \mathcal{D}]}_{\textit{Model Uncertainty}} = \underbrace{\mathcal{H}[\mathbb{E}_{\mathrm{p}(\boldsymbol{\theta}|\mathcal{D})}[\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta})]]}_{\textit{Total Uncertainty}} - \underbrace{\mathbb{E}_{\mathrm{p}(\boldsymbol{\theta}|\mathcal{D})}[\mathcal{H}[\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta})]]}_{\textit{Expected Data Uncertainty}}$$

- Prior Network uncertainty decomposition

$$\underbrace{\mathcal{I}[y, \boldsymbol{\mu} | \boldsymbol{x}^*; \hat{\boldsymbol{\theta}}]}_{\textit{Knowledge Uncertainty}} = \underbrace{\mathcal{H}\big[\mathbb{E}_{\mathrm{p}(\boldsymbol{\mu}|\boldsymbol{x}^*; \hat{\boldsymbol{\theta}})}[\mathrm{P}(y|\boldsymbol{\mu})]\big]}_{\textit{Total Uncertainty}} - \underbrace{\mathbb{E}_{\mathrm{p}(\boldsymbol{\mu}|\boldsymbol{x}^*; \hat{\boldsymbol{\theta}})}\big[\mathcal{H}[\mathrm{P}(y|\boldsymbol{\mu})]\big]}_{\textit{Expected Data Uncertainty}}$$

Prior Networks vs. Ensembles [Malinin and Gales, 2018]

- Behaviour of Ensemble distribution over distributions
  - Controlled via prior $\mathrm{p}(\boldsymbol{\theta})$ and inference scheme
- Behaviour of Prior Networks distribution over distributions
  - Controlled via loss function and training data $\mathcal{D}$

Prior Network Construction [Malinin and Gales, 2018]

- A Prior Network parametrizes the **Dirichlet Distribution**

$$\mathrm{p}(\boldsymbol{\mu}|\boldsymbol{x}^*;\hat{\boldsymbol{\theta}}) = \mathtt{Dir}(\boldsymbol{\mu}|\boldsymbol{\alpha}), \quad \boldsymbol{\alpha} = \boldsymbol{f}(\boldsymbol{x}^*;\hat{\boldsymbol{\theta}})$$

- Dirichlet Distribution $\rightarrow$ Distribution over simplex
  - Conjugate prior to categorical distribution
  - Convenient properties $\rightarrow$ analytically tractable

Reminder - Dirichlet Distribution

- Dirichlet is a distribution over categorical distributions

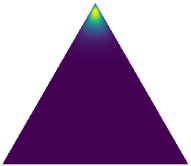
$$\texttt{Dir}(\boldsymbol{\mu}|\boldsymbol{\alpha}) = \frac{\Gamma(\alpha_0)}{\prod_{c=1}^{K}\Gamma(\alpha_c)}\prod_{c=1}^{K}\pi_c^{\alpha_c-1}; \quad \alpha_0 = \sum_{c=1}^{K}\alpha_c$$

  - Parameterised by concentration parameters: $\boldsymbol{\alpha}$, $\alpha_c > 0$
  - Expected label posteriors given by

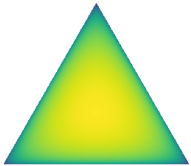
$$\hat{\mathrm{P}}(y=\omega_c) = \hat{\mu}_c = \frac{\alpha_c}{\sum_{k=1}^{K}\alpha_k}$$

Prior Network Construction [Malinin and Gales, 2018]

$$\mathcal{L}(\boldsymbol{\theta}, \mathcal{D}) = \underbrace{\mathcal{L}_{in}(\boldsymbol{\theta}, \mathcal{D}_{\mathtt{trn}})}_{\textit{In Domain Loss}} + \gamma \cdot \underbrace{\mathcal{L}_{out}(\boldsymbol{\theta}, \mathcal{D}_{\mathtt{out}})}_{\textit{OOD Loss}}$$



(a) In-Domain Target



(b) OOD Target

Target Concentration Parameters [Malinin and Gales, 2018]

- To train the prior network we need a target distribution $\mathrm{p}(\boldsymbol{\mu}|\boldsymbol{\beta})$ for $\boldsymbol{x}^{(i)}$
  - We **want** training data $\{\boldsymbol{\beta}^{(i)}, \boldsymbol{x}^{(i)}\}_{i=1}^{N}$
  - ... but have training data $\{y^{(i)}, \boldsymbol{x}^{(i)}\}_{i=1}^{N}$, where $y^{(i)} \in \{\omega_1, \ldots, \omega_K\}$

- Solution $\rightarrow$ specify concentration parameters $\boldsymbol{\beta}^{(c)}$ as a function of target class y

$$\mathrm{p}(\boldsymbol{\mu}|\boldsymbol{\beta}^{(c)}) = \mathtt{Dir}(\boldsymbol{\mu}|\boldsymbol{\beta}^{(c)})$$

  - Need $\boldsymbol{\beta}^{(c)}$ to yield correct class
  - Need $\boldsymbol{\beta}^{(c)}$ to reflect "confidence" in sample
  - $\beta_k > 0 \ \forall k$

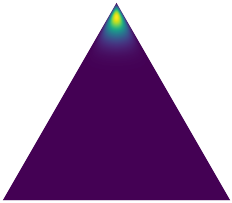
Target Concentration Parameters [Malinin and Gales, 2018]

- Consider setting $\boldsymbol{\beta}^{(c)}$ as follows $\rightarrow$

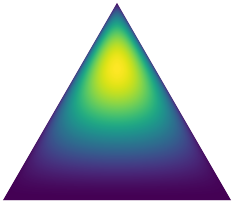
$$\beta_k^{(c)} = \begin{cases} \beta + 1 & \text{if } c = k \\ 1 & \text{if } c \neq k \end{cases}$$

- If β is large $\rightarrow$
  - Sharp Dirichlet at corner of simplex corresponding to target class.
- If β is low $\rightarrow$
  - Wide Dirichlet with the mode near the corner corresponding to target class.
- If β is zero $\rightarrow$
  - Flat (uniform) Dirichlet distribution.

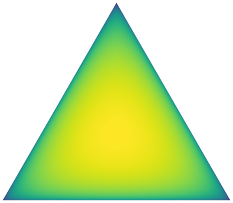
Target Concentration Parameters [Malinin and Gales, 2018]



(c) $\beta = 30$



(d) $\beta = 2$



(e) $\beta = 0$

KL-divergence Losses

- We can consider two loss functions - *Forward KL-Divergence* $\rightarrow$

$$\mathcal{L}^{KL}(y, \boldsymbol{x}, \boldsymbol{\theta}; \beta) = \sum_{c=1}^{K} \mathcal{I}(y = \omega_c) \cdot \mathtt{KL}[\mathtt{Dir}(\boldsymbol{\mu}|\boldsymbol{\beta}^{(c)})||\mathtt{p}(\boldsymbol{\mu}|\boldsymbol{x}; \boldsymbol{\theta})]$$

- ... or *reverse KL-Divergence* $\rightarrow$

$$\mathcal{L}^{RKL}(y, \boldsymbol{x}, \boldsymbol{\theta}; \beta) = \sum_{c=1}^{K} \mathcal{I}(y = \omega_c) \cdot \mathtt{KL}[\mathtt{p}(\boldsymbol{\mu}|\boldsymbol{x}; \boldsymbol{\theta})||\mathtt{Dir}(\boldsymbol{\mu}|\boldsymbol{\beta}^{(c)})]$$

Reminder - Kullback-Leibler Divergence

- Standard "measure" between distributions

$$\mathrm{KL}[\mathrm{P}_{\mathrm{tr}}(y|\boldsymbol{x})||\mathrm{P}(y|\boldsymbol{x};\boldsymbol{\theta})] = \mathbb{E}_{\mathrm{P}_{\mathrm{tr}}(y|\boldsymbol{x})}\left[\ln \mathrm{P}_{\mathrm{tr}}(y|\boldsymbol{x}) - \ln \mathrm{P}(y|\boldsymbol{x};\boldsymbol{\theta})\right]$$

- Variational optimization often yields reverse KL for training

$$\mathrm{KL}[\mathrm{P}(y|\boldsymbol{x};\boldsymbol{\theta})||\mathrm{P}_{\mathrm{tr}}(y|\boldsymbol{x})] = \mathbb{E}_{\mathrm{P}(y|\boldsymbol{x};\boldsymbol{\theta})}\left[\ln \mathrm{P}(y|\boldsymbol{x};\boldsymbol{\theta}) - \ln \mathrm{P}_{\mathrm{tr}}(y|\boldsymbol{x})\right]$$

- Measures have different properties $\rightarrow$
  - Forward KL is **zero-avoiding**
  - Reverse KL is zero-forcing

Forward KL-divergence Loss [Malinin and Gales, 2018]

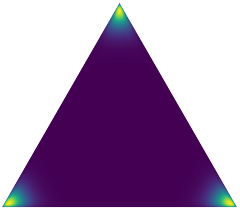
- Consider expectation of *forward* KL-div loss wrt. empirical distribution $\hat{\mathrm{p}}(\boldsymbol{x}, y) \rightarrow$

$$\begin{aligned}
\mathcal{L}^{KL}(\boldsymbol{\theta}; \beta) &= \mathbb{E}_{\hat{\mathrm{p}}_{\mathrm{tr}}(\boldsymbol{x},y)}\Big[\sum_{c=1}^{K} \mathcal{I}(y = \omega_c) \cdot \mathtt{KL}[\mathtt{Dir}(\boldsymbol{\mu}|\boldsymbol{\beta}^{(c)})||\mathrm{p}(\boldsymbol{\mu}|\boldsymbol{x}; \boldsymbol{\theta})]\Big] \\
&= \mathbb{E}_{\hat{\mathrm{p}}_{\mathrm{tr}}(\boldsymbol{x})}\Big[\sum_{c=1}^{K} \mathbb{E}_{\hat{\mathrm{p}}_{\mathrm{tr}}(y|\boldsymbol{x})}[\mathcal{I}(y = \omega_c)] \cdot \mathtt{KL}[\mathtt{Dir}(\boldsymbol{\mu}|\boldsymbol{\beta}^{(c)})||\mathrm{p}(\boldsymbol{\mu}|\boldsymbol{x}; \boldsymbol{\theta})]\Big] \\
&= \mathbb{E}_{\hat{\mathrm{p}}_{\mathrm{tr}}(\boldsymbol{x})}\Big[\mathtt{KL}[\sum_{c=1}^{K} \mathrm{P}_{\mathrm{tr}}(y = \omega_c) \cdot \mathtt{Dir}(\boldsymbol{\mu}|\boldsymbol{\beta}^{(c)})||\mathrm{p}(\boldsymbol{\mu}|\boldsymbol{x}; \boldsymbol{\theta})]\Big] + C
\end{aligned}$$

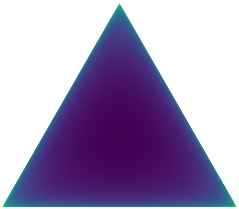
- Target distribution becomes a **mixture of Dirichlets**!

Forward KL-divergence Loss [Malinin and Gales, 2018]

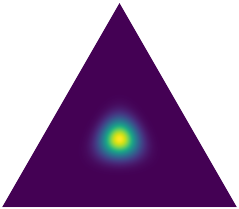
- Target distribution becomes a **mixture of Dirichlets**!



(a) Indueced Target



(b) Model



(c) Want

- Forward KL-divergence is zero avoiding $\rightarrow$ Model will try to cover **each mode**!
  - Leads to **undesired behaviour** $\rightarrow$ bad performance!
  - **Doesn't scale** to datasets with more than 10 classes!

Reverse KL-divergence Loss [Malinin and Gales, 2019]

- Consider expectation of *reverse* KL-div loss wrt. empirical distribution $\hat{\mathrm{p}}(\boldsymbol{x}, y) \rightarrow$

$$\begin{aligned}
\mathcal{L}^{RKL}(\boldsymbol{\theta}; \beta) &= \mathbb{E}_{\hat{\mathrm{p}}_{\mathtt{tr}}(\boldsymbol{x})}\Big[\sum_{c=1}^{K} \hat{\mathrm{P}}_{\mathtt{tr}}(y = \omega_c|\boldsymbol{x})\mathtt{KL}\big[\mathrm{p}(\boldsymbol{\mu}|\boldsymbol{x};\boldsymbol{\theta})||\mathtt{Dir}(\boldsymbol{\mu}|\boldsymbol{\beta}^{(c)})\big]\Big] \\
&= \mathbb{E}_{\hat{\mathrm{p}}_{\mathtt{tr}}(\boldsymbol{x})}\Big[\mathbb{E}_{\mathrm{p}(\boldsymbol{\mu}|\boldsymbol{x};\boldsymbol{\theta})}\big[\ln \mathrm{p}(\boldsymbol{\mu}|\boldsymbol{x};\boldsymbol{\theta}) - \ln \prod_{c=1}^{K} \mathtt{Dir}(\boldsymbol{\mu}|\boldsymbol{\beta}^{(c)})^{\hat{\mathrm{P}}_{\mathtt{tr}}(y=\omega_c|\boldsymbol{x})}\big]\Big] \\
&= \mathbb{E}_{\hat{\mathrm{p}}_{\mathtt{tr}}(\boldsymbol{x})}\Big[\mathtt{KL}\big[\mathrm{p}(\boldsymbol{\mu}|\boldsymbol{x};\boldsymbol{\theta})||\mathtt{Dir}(\boldsymbol{\mu}|\sum_{c=1}^{K} \hat{\mathrm{P}}_{\mathtt{tr}}(y = \omega_c|\boldsymbol{x}) \cdot \boldsymbol{\beta}^{(c)})\big]\Big] + C
\end{aligned}$$

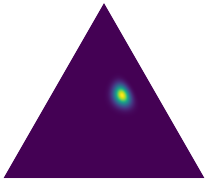
- Expectation induces product of target Dirichlet distributions.

Reverse KL-divergence Loss [Malinin and Gales, 2019]

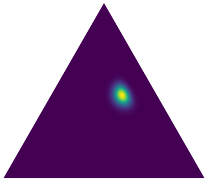
- Expectation induces product of target Dirichlet distributions

$$\mathcal{L}^{RKL}(\boldsymbol{\theta};\beta) = \mathbb{E}_{\hat{\mathrm{p}}_{\mathrm{tr}}(\boldsymbol{x})}\Big[\mathrm{KL}\big[\mathrm{p}(\boldsymbol{\mu}|\boldsymbol{x};\boldsymbol{\theta})||\mathrm{Dir}(\boldsymbol{\mu}|\sum_{c=1}^{K}\hat{\mathrm{P}}_{\mathrm{tr}}(y=\omega_c|\boldsymbol{x})\cdot\boldsymbol{\beta}^{(c)})\big]\Big] + C$$

- Target becomes a **uni-modal Dirichlet distribution** at appropriate location!



(a) Induced Target



(b) Model

Prior Network Construction

- Reverse KL loss $\mathcal{L}^{RKL}(\boldsymbol{\theta}, \mathcal{D}; \beta) \rightarrow$ full control over behaviour of model!

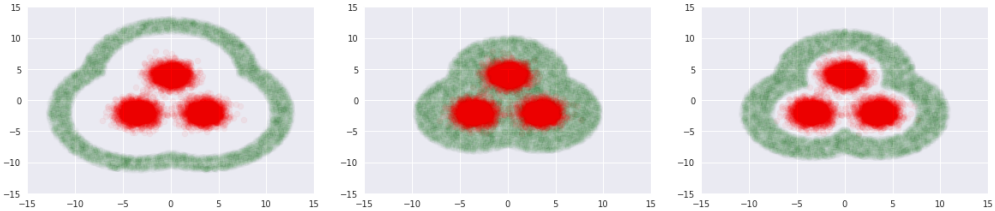
$$\mathcal{L}(\boldsymbol{\theta}, \mathcal{D}; \beta_{in}, \beta_{out}, \gamma) = \mathcal{L}^{RKL}_{in}(\boldsymbol{\theta}, \mathcal{D}_{trn}; \beta_{in}) + \gamma \cdot \mathcal{L}^{RKL}_{out}(\boldsymbol{\theta}, \mathcal{D}_{out}; \beta_{out})$$

(a) In-Domain Target (b) OOD Target

- But how to obtain out-of-domain training data $\mathcal{D}_{out}$?
  - Use a different dataset or **adversarial attacks**

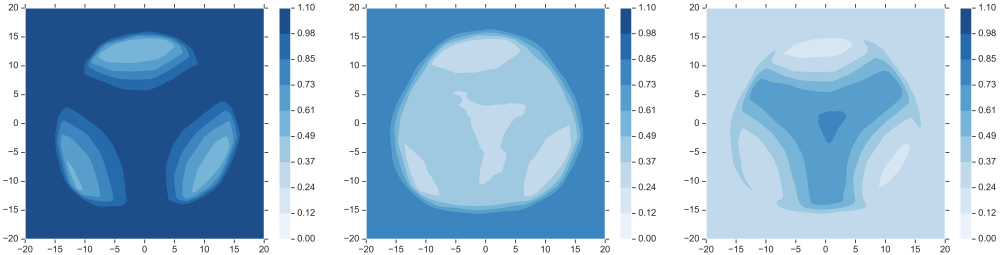
Prior Network Construction

- Out-of-domain (OOD) training data must be on *boundary* on in-domain region $\rightarrow$
  - Too loose $\rightarrow$ Some OOD might be considered in-domain
  - Too tight $\rightarrow$ Some in-domain might be considered OOD



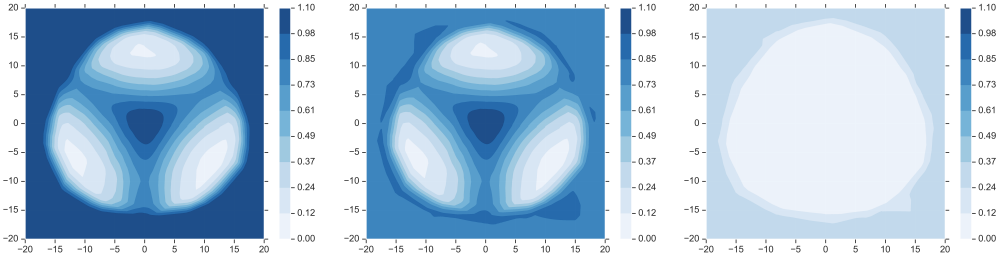
(a) Too Loose (b) Too Tight (c) Good

Prior Networks trained with *forward* KL-divergence loss on Artificial Data



(a) Total Uncertainty (b) Data Uncertainty (c) Knowledge Uncertainty

Prior Networks trained with *reverse* KL-divergence loss on Artificial Data



(a) Total Uncertainty (b) Data Uncertainty (c) Knowledge Uncertainty

Classification Error Rate

Dataset	DNN	PN-KL	PN-RKL	Ensemble
CIFAR-10	8.0	14.7	7.5	**6.6**
CIFAR-100	30.4	-	28.1	**26.9**
TinyImageNet	41.7	-	40.3	**36.9**

Out-of-Distribution Detection

Model	CIFAR-10/CIFAR-100			CIFAR-100/TinyImageNet		
	SVHN	LSUN	TinyImageNet	SVHN	LSUN	CIFAR-10
Ensemble	89.5	93.2	90.3	78.9	85.6	**76.5**
PN-KL	97.8	91.6	92.4	-	-	-
PN-RKL	**98.2**	**95.7**	**95.7**	**84.8**	**100.0**	57.8

Table: Out-of-domain detection results (mean % AUROC across 10 rand. inits).

Overview of the Talk

1. Context: Why do we need Uncertainty Estimation?
2. Sources of Uncertainty in Predictions
3. Ensemble Approaches
4. Prior Networks
5. **Adversarial Attack Detection**

Adversarial Attack Detection

- Adversarial Attacks $\rightarrow$ Small perturbation of the input $\boldsymbol{x}^*$ which affects prediction
  - Exist for **many modalities** $\rightarrow$ images, text and audio
  - Transferable between models
  - Can be deployed in **real world**.
- Adversarial Attacks are a **security concern**!
  - Detect using measures of **uncertainty**?

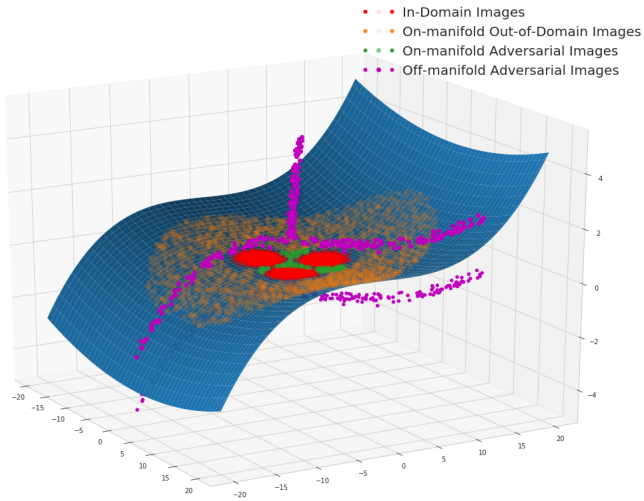
Adversarial Attack Generation

- Adversarial attacks: generate sample $\tilde{\boldsymbol{x}}$:
  1. swaps to target class $\tilde{\omega}$
  2. requires minimum changes to original sample $\boldsymbol{x}$
- Requirements expressed as

$$\mathcal{A}_{\text{adv}}(\boldsymbol{x}, \tilde{\omega}) = \arg \min_{\tilde{\boldsymbol{x}} \in \mathcal{R}^D} \left\{ \mathcal{L}(y = \tilde{\omega}, \tilde{\boldsymbol{x}}, \hat{\boldsymbol{\theta}}) \right\} : \ \delta(\boldsymbol{x}, \tilde{\boldsymbol{x}}) < \epsilon$$

- ϵ is number of swapped bits (for images)

Manifold Interpretation of Adversarial Attacks



Adversarial Attack Detection

- Consider an uncertainty based detection scheme:

$$\hat{\mathcal{I}}_T(\boldsymbol{x}) = \begin{cases} 1, & T > \mathcal{H}(\boldsymbol{x}) \\ 0, & T \leq \mathcal{H}(\boldsymbol{x}) \\ 0, & \boldsymbol{x} = \emptyset \end{cases}$$

- Successful attacks are able to :
  - Both **affect prediction** and avoid detection.
- Can assess using false positive and true positives:

$$t_p(T) = \frac{1}{N}\sum_{i=1}^{N} \mathcal{I}_T(\boldsymbol{x}_i), \quad f_p(T) = \frac{1}{N}\sum_{i=1}^{N} \mathcal{I}_T(\mathcal{A}_{\text{adv}}(\boldsymbol{x}_i, \omega_t))$$

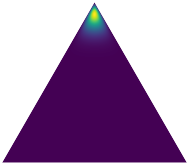
- **Joint Success Rate** is where $t_p(T) = f_p(T)$

Adversarial Attack Detection via Prior Networks

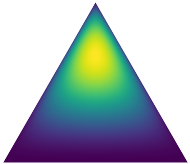
- Prior Networks yield rich measures of uncertainty
- Greatly constrain space of successful *adaptive* attacks
  - Confidence $\rightarrow$ constraint on max logit
  - Total Uncertainty $\rightarrow$ constraint on relative of magnitude of logits
  - Knowledge Uncertainty $\rightarrow$ constraint on relative and absolute magnitude of logits
  - .. and attack must also affect predicted class!
- Explicit control behaviour via training data $\rightarrow$
  - Further constrain space of successful adversarial solutions

Prior Network Adversarial Training

$$\mathcal{L}(\boldsymbol{\theta}, \mathcal{D};\ \beta_{nat}, \beta_{adv}) = \mathcal{L}^{RKL}_{nat}(\boldsymbol{\theta}, \mathcal{D}_{\texttt{trn}}; \beta_{\texttt{nat}} = 1e2) + \gamma \cdot \mathcal{L}^{RKL}_{\texttt{adv}}(\boldsymbol{\theta}, \mathcal{D}_{\texttt{adv}}; \beta_{\texttt{adv}} = 1)$$



(a) Natural Target



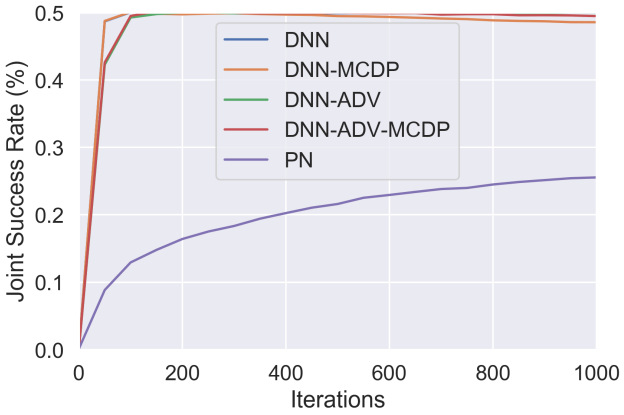
(b) Adversarial Target

- Standard adversarial training $\rightarrow$
  - Correct Prediction for adversarial inputs
- Prior Network adversarial training $\rightarrow$
  - Correct Prediction + High Uncertainty for adversarial inputs

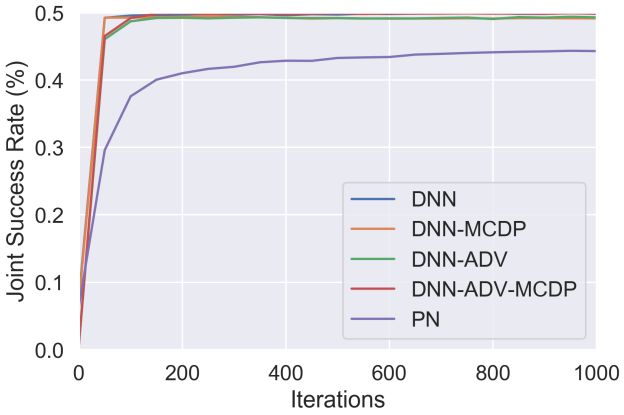
Experiments

- Baselines: MC-Dropout and Standard Adversarial Training
- Prior Network adversarial traning $\rightarrow$ 1-step FGSM attacks
- Evaluation Attack $\rightarrow$ strong *adaptive* whitebox PGD-MIM attack
- Datasets: CIFAR10 and CIFAR100

Joint Success Rate - CIFAR10



Joint Success Rate - CIFAR100



Thank You!

Any questions?

References I

[Blundell et al., 2015] Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. (2015).
Weight uncertainty in neural networks.
arXiv preprint arXiv:1505.05424.

[Gal, 2016] Gal, Y. (2016).
Uncertainty in Deep Learning.
PhD thesis, University of Cambridge.

[Gal and Ghahramani, 2016] Gal, Y. and Ghahramani, Z. (2016).
Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning.
In *Proc. 33rd International Conference on Machine Learning (ICML-16)*.

References II

[Garipov et al., 2018] Garipov, T., Izmailov, P., Podoprikhin, D., Vetrov, D. P., and Wilson, A. G. (2018).
Loss surfaces, mode connectivity, and fast ensembling of dnns.
In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 31*, pages 8789–8798. Curran Associates, Inc.

[Hendrycks and Gimpel, 2016] Hendrycks, D. and Gimpel, K. (2016).
A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks.
http://arxiv.org/abs/1610.02136.
arXiv:1610.02136.

References III

[Hernández-Lobato and Adams, 2015] Hernández-Lobato, J. M. and Adams, R. (2015).
Probabilistic backpropagation for scalable learning of bayesian neural networks.
In *International Conference on Machine Learning*, pages 1861–1869.

[Lakshminarayanan et al., 2017] Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017).
Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles.
In *Proc. Conference on Neural Information Processing Systems (NIPS)*.

[Maddox et al., 2019] Maddox, W., Garipov, T., Izmailov, P., Vetrov, D. P., and Wilson, A. G. (2019).
A simple baseline for bayesian uncertainty in deep learning.
CoRR, abs/1902.02476.

References IV

[Malinin and Gales, 2018] Malinin, A. and Gales, M. (2018).
Predictive uncertainty estimation via prior networks.
In *Advances in Neural Information Processing Systems*, pages 7047–7058.

[Malinin and Gales, 2019] Malinin, A. and Gales, M. (2019).
Reverse kl-divergence training of prior networks: Improved uncertainty and adversarial robustness.
arXiv preprint arXiv:1905.13472.

[Osband et al., 2016] Osband, I., Blundell, C., Pritzel, A., and Van Roy, B. (2016).
Deep exploration via bootstrapped dqn.
In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems 29*, pages 4026–4034. Curran Associates, Inc.

References V

[Welling and Teh, 2011] Welling, M. and Teh, Y. W. (2011).
Bayesian Learning via Stochastic Gradient Langevin Dynamics.
In *Proc. International Conference on Machine Learning (ICML)*.