Increasing Trustworthiness of Deep Neural Networks (DNNs) via Accuracy Monitoring

Zhihui Shao, Jianyi Yang, and Shaolei Ren

UC Riverside

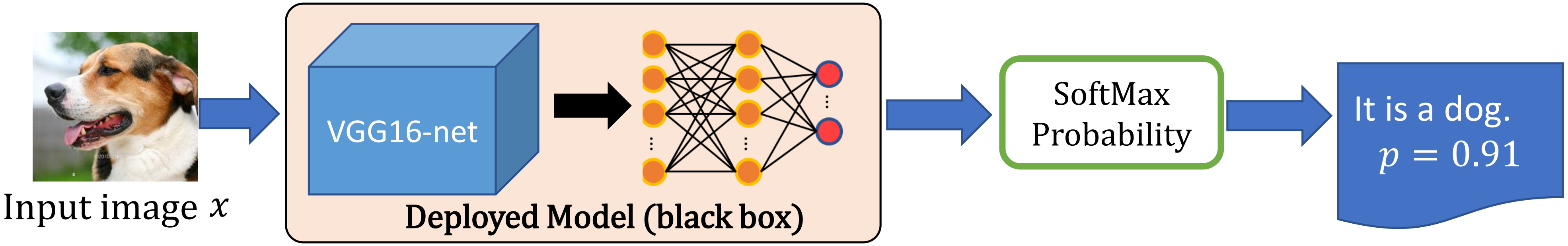
Acknowledgement: This work was supported by the U.S. NSF under grants CNS-1551661, ECCS-1610471, and CNS-1910208.

Outline

- Trustworthiness for DNN
- DNN Accuracy Monitor
- Experiment
- Conclusion

Trustworthiness for DNN - background

➢ **DNN inference process**



➢ **Trustworthiness**

Problem:

- How to obtain DNN accuracy on the user dataset D^U?
  - Easy for dataset with labels.
- Estimate/monitor accuracy with (x_i, p_i).

$$Acc = \frac{1}{|\mathcal{D}^U|} \sum_{(x_i, y_i) \in \mathcal{D}^U} \mathbf{I}(y_i = \tilde{y}_i)$$

Challenges:

1. **In real-world, we don't have labels!**
   - Or user dataset with limited labels.
2. Black-box model for deployed DNN.
   - DNNs are provided by other vendors,
     - Machine learning as a service (MLaaS).

Trustworthiness for DNN – related works

- **Uncertainty estimation**

Target: estimate the uncertainty/accuracy for each input data.

1. Train DNN with ensemble models
- But model are provided by vendor, cannot be re-trained.

2. Re-sample the DNN model weights
- But weights of black-box DNN are not accessible.

3. Estimate by the similarity between training and test.
- But training dataset is unknown.

- Requires training knowledge of deployed model!

Orthogonal to our work!

Trustworthiness for DNN – related works

- **Direct accuracy estimation**

Target: directly estimate accuracy of DNN

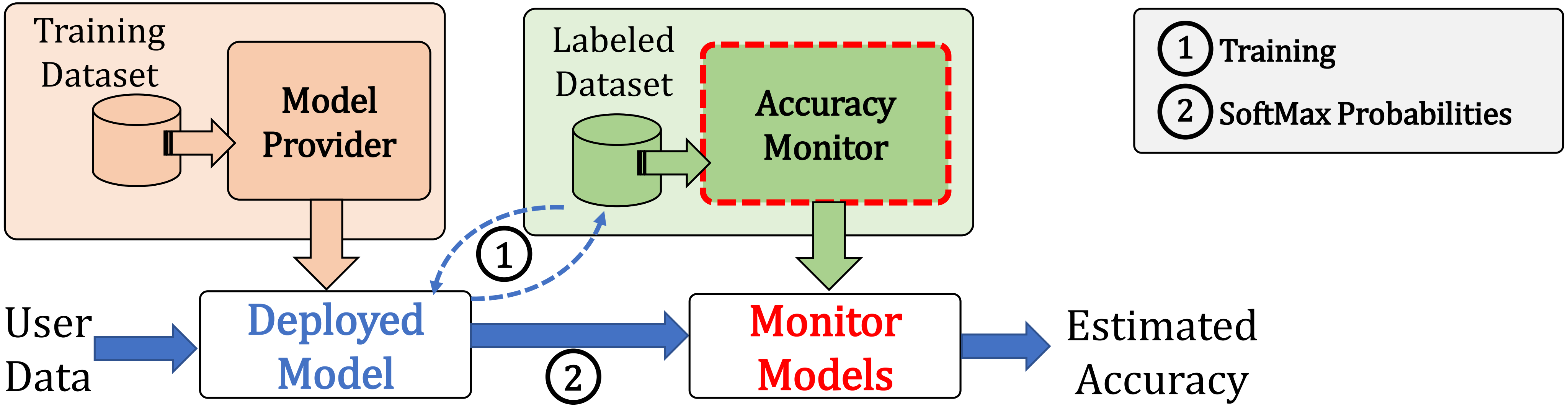
1. Label with **random sampling**
   - label $u\%$ user data, and then estimate accuracy with these labels.
2. **SoftMax** probability method
3. **Entropy** method
4. **Temperature scaling** method
   - Re-calculate the SoftMax probability: $$p_T = \frac{\mathrm{e}^{y_i/T}}{\sum_{k=1}^{\mathrm{N}} \mathrm{e}^{y_k/T}}$$

Consider as baselines in our work!

Outline

- Trustworthiness for DNN
- DNN Accuracy Monitor
- Experiment
- Conclusion

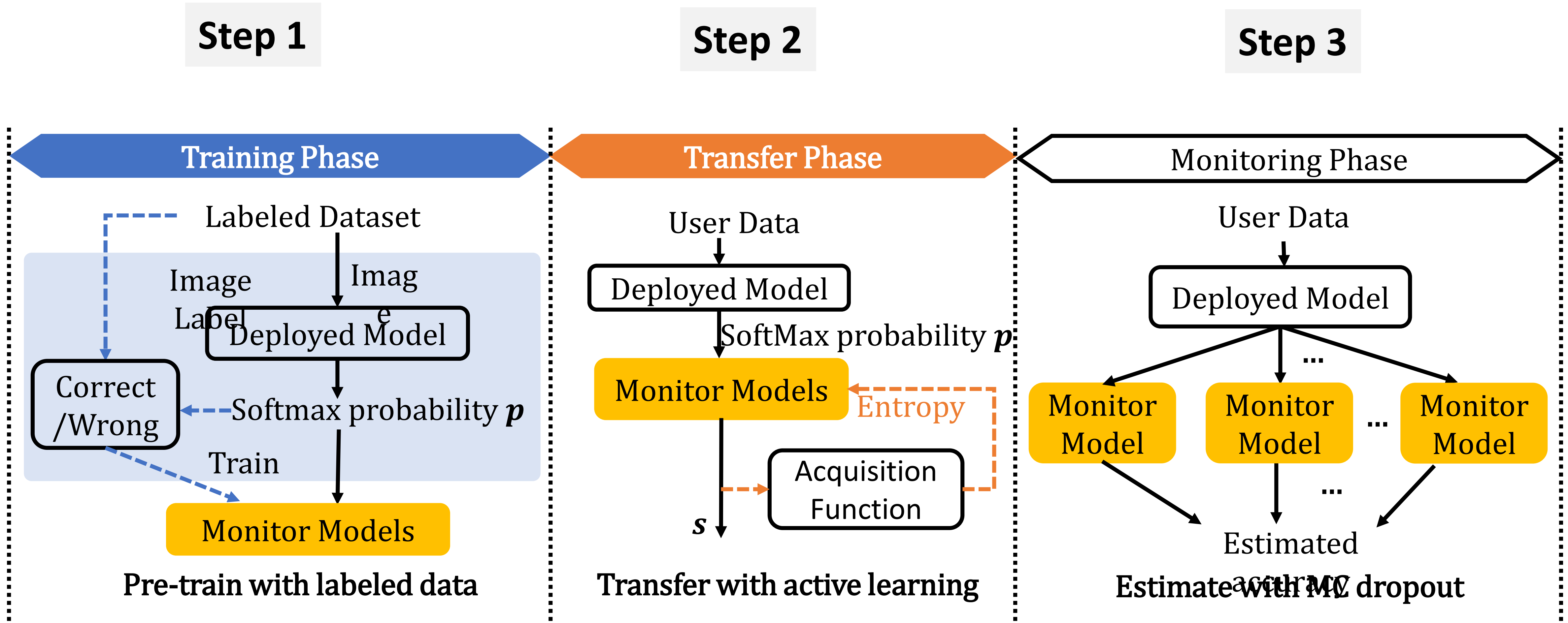
DNN Accuracy Monitor – illustration



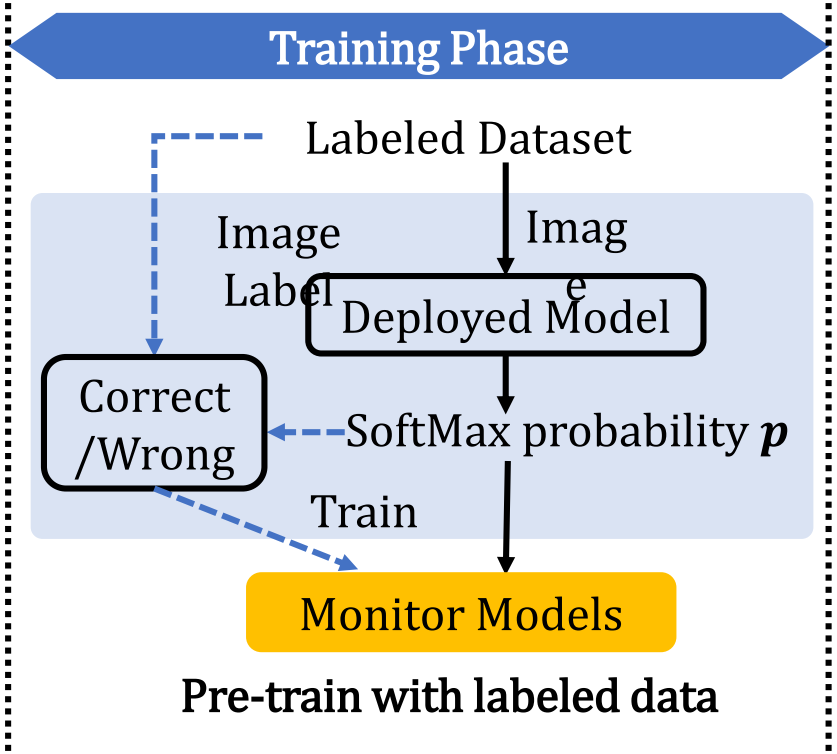
- **Deployed model: (Given)**
  - Provide by product vendor.
    - Input: user data x;
    - Output: SoftMax probability p;
  - Black-box model for user.

- **Monitor models: (our method)**
  - Add-on part developed by third-party auditor.
    - Input: inference probability p;
    - Output: estimated accuracy;
  - Shallow models (MLP) with binary output.

DNN Accuracy Monitor – Monitor Learning



DNN Accuracy Monitor – Model Training

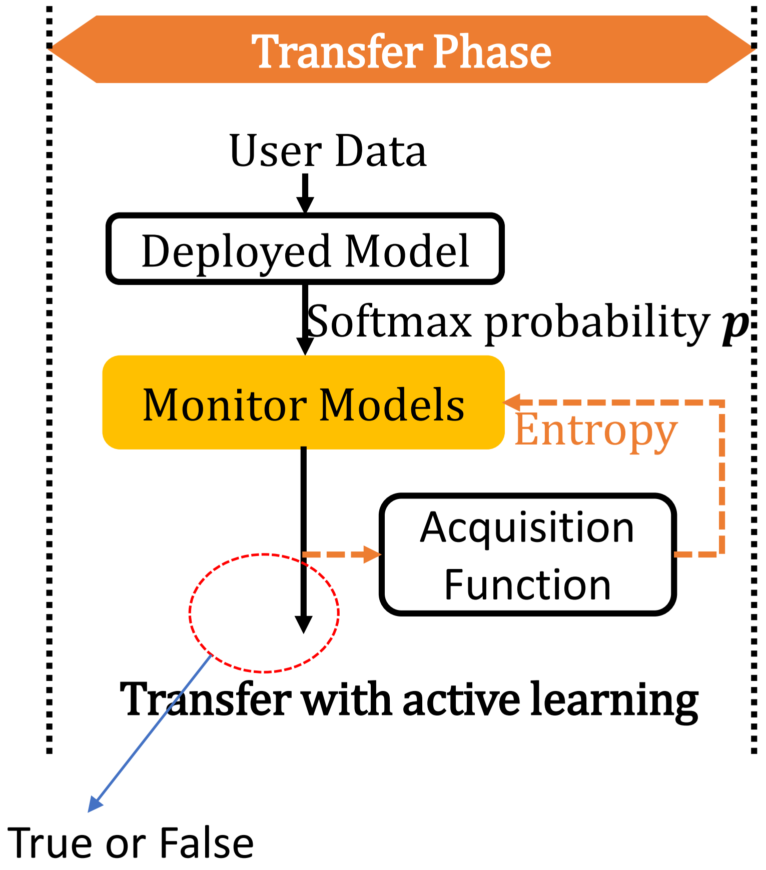


- Pre-training with labeled data:
  - Use public dataset (known labels).
  - Generate probability p on deployed model.
  - Obtain Correct/Wrong (CW) label of deployed model's estimation.
  - Train monitor on data pair (p, CW).
- Loss function (binary cross-entropy):

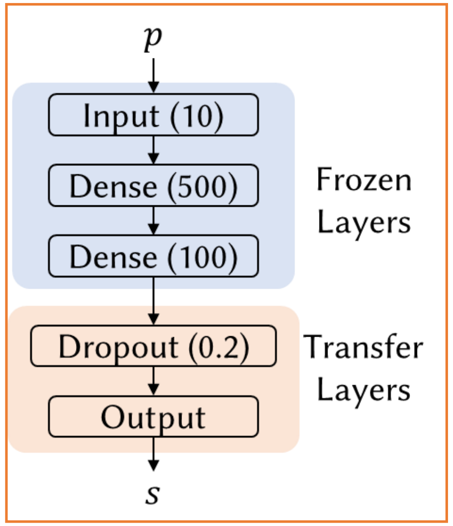
$$Loss = -\frac{1}{N}\sum_{i=1}^{N} CW_i \cdot \log M_\Theta(p) + (1 - CW_i) \cdot \log(1 - M_\Theta(p))$$

- Ensembled monitors:
  - Achieve robust estimation.

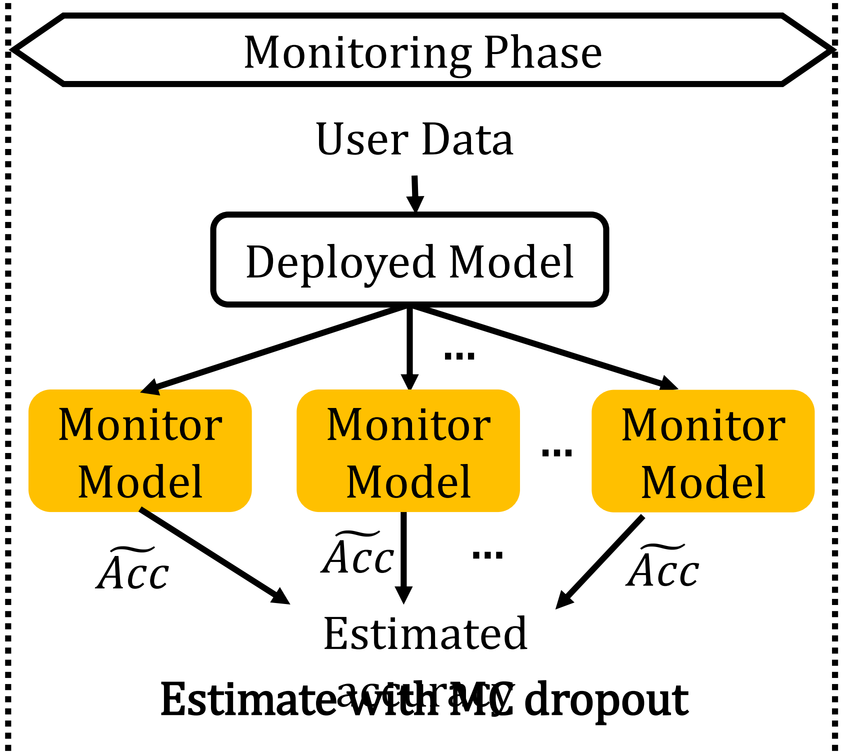
DNN Accuracy Monitor – Model Transfer



- Model transfer with active learning:
  - Generate probability p for D^t.
  - Actively label $t\%$ sample from target dataset D^t.
    - Train monitors on labeled data.
    - Label data with a higher entropy.
- Transfer learning
  - Frozen the prior layers.
  - Transfer on the output layer.



DNN Accuracy Monitor – Monitoring



Estimate with MC dropout

- DNN monitoring with ensembled models:
  - Generate probability p for user data D^U.
  - Predict correctness probability s from each monitor model.
    - MC dropout method is used for monitoring phase.
  - Model accuracy is estimated from correctness probability s with threshold th_s.

$$\widetilde{Acc} = \frac{1}{|D^U|} \sum_{x \in D^U} \mathbb{I}[s(p_x) \geq th_s]$$

  - Ensembled models provide a more robust estimation.

Outline

- Trustworthiness for DNN
- DNN Accuracy Monitor
- Experiment
- Conclusion

DNN Accuracy Monitor – Experiment

- **Setup**

  Application:
  - General image classification
    - small-scale image – CIFAR-10
    - large-scale image – ImageNet
  - Traffic sign detection for autonomous driving

  Models and user dataset:
  - small-scale image (10 class)
    - Model: VGG16; Dataset: CIFAR-10, CINIC-10, STL-10, AD-10
  - large-scale image (1000 class)
    - Model: ResNet-50 and MobileNet; Dataset: ImageNet Validation
  - Traffic sign detection (43 class)
    - Model: AlexNet; Dataset: German Traffic Sign Detection (GTSD)

Result – CIFAR-10

- **Summary**
  - True inference accuracy of VGG16 model:
    - 0.9356 (CIFAR-10), 0.7617 (CINIC-10), 0.6304 (STL-10), and 0.3780 (AD-10).
  - Our method provides more accurate estimation.
    - The std is provided by ensembled monitors for estimated accuracy.
    - Better than RS with 10x labelled data.

Method	Estimated Accuracy			
	CIFAR-10	CINIC-10	STL-10	AD-10
Our method	**0.9313**/0.0123	**0.7691**/0.0138	**0.6343**/0.0371	**0.3866**/0.0322
MP	0.8907	0.7574	0.7105	0.5035
Entropy	0.8943	0.7662	0.7165	0.5380
TS	0.9727	0.4066	0.8803	0.8618
MP*	0.9756	0.9443	0.9319	0.7881
RS (1%)	[0.8879,0.9852]	[0.6500,0.7340]	[0.5274,0.7382]	[0.2800,0.5100]
RS (10%)	[0.9207,0.9516]	[0.7340,0.7930]	[0.5976,0.6618]	[0.3400,0.4080]

mean/std

Result – ImageNet

- **Summary**
  - True inference accuracy of model:
    - MobileNet: 0.6859 (ImageNet-A), 0.6791 (ImageNet-B)
    - ResNet-50: 0.6836 (ImageNet-A), 0.6727 (ImageNet-B)
  - Our method still provides better result.
    - With similar distribution, the SoftMax can provide a good accuracy estimation.

Method	Estimated Accuracy			
	MobileNet		ResNet-50	
	ImageNet A	ImageNet B	ImageNet A	ImageNet B
Our method	**0.6933**/0.0202	**0.6796**/0.0235	**0.6862**/0.0240	**0.6719**/0.0219
MP	0.7203	0.7004	0.6765	0.6757
Entropy	0.7032	0.7131	0.6724	0.6694
TS	0.8094	0.8086	0.7771	0.8044
MP*	0.7550	0.7539	0.7633	0.7638
RS (1%)	[0.6197,0.7512]	[0.5866,0.7754]	[0.6631,0.7029]	[0.6457,0.7049]
RS (10%)	[0.6652,0.7057]	[0.6492,0.7073]	[0.6696,0.6987]	[0.6525,0.6959]

mean/std

Result – Traffic Sign Detection

- **Summary**
  - GTSD (German Traffic Sign Detection)
    - Generate 4 user datasets:
      - D1 (original), D2 (spatial transformation), OOD (~50% images from CIFAR-10)
      - AD (~ 50% adversarial images)
    - True accuracy: D1(0.9734), D2(0.8401), OOD(0.5147), AD(0.4291)
  - Our method still provides better result.

Method	Estimated Accuracy			
	GTSD-D1	GTSD-D2	GTSD-OOD	GTSD-AD
Our method	**0.9735**/0.001	**0.8414**/0.005	**0.5362**/0.005	**0.4162**/0.001
MP	0.9837	0.7991	0.5690	0.4886
Entropy	0.9621	0.7866	0.5821	0.4806
TS	0.9855	0.8004	0.7157	0.6884
MP*	0.9895	0.9390	0.8861	0.9176
RS (1%)	[0.9406,1.000]	[0.7624,0.9307]	[0.4300,0.5900]	[0.3533,0.4900]
RS (10%)	[0.9574,0.9871]	[0.8178,0.8693]	[0.4880,0.5387]	[0.4100,0.4460]

Outline

- Trustworthiness for DNN
- DNN Accuracy Monitor
- Experiment
- Conclusion

Conclusion

- **Develop a monitor model to estimate DNN's inference accuracy**
  - Our method is applicable to black-box model
    - No need to re-train the deployed model
    - No prior information required for the deployed model
  - Monitor model serves as a **plug-in module**
    - Evaluate the DNN performance
  - Ensembled method (MC dropout) + active learning
    - Less labelled data required compared with randomly labelling.

Thank you!

- Please contact us with questions and comments.
  - Zhihui Shao (*zshao006@ucr.edu*)
  - Jianyi Yang (*jyang239@ucr.edu*)
  - Shaolei Ren (*sren@ece.ucr.edu*)

Appendix - Baselines

- **Random sampling (RS)**
  - $u\%$ of user's data is randomly sampled and manually labeled.
  - Accuracy on the labeled samples is considered as the overall accuracy.
- **Maximum probability (MP)**
  - Maximum SoftMax probability: $MP = \max_{k \in \{1,2,..C\}} \{p_k(x)\}$
  - **MP***:
    - no manual labeling is required.
    - Average Maximum SoftMax probability is the estimated accuracy.
  - **MP**:
    - $MP(x) \geq th_{MP}$ is considered as correct. (th_{MP} from labelled data)
- **Entropy:**
  - $Entropy(x) \leq th_{En}$ is considered as correct. (th_{En} from labelled data)
- **Temperature scaling (TS):**
  - Re-calibrate SoftMax probability and then average p_T.
  - T is obtained from labelled data

$$p_T = \frac{e^{y_i/T}}{\sum_{k=1}^{N} e^{y_k/T}}$$

Appendix – Training Algorithms

Algorithm 1: DNN Accuracy Monitoring

Input: A labeled dataset $\mathcal{D}^R$, user dataset $\mathcal{D}^U$, target model $M_{\Theta_d}(x)$, the MC dropout model number B, data labeling budget $t\%$.

1. Obtain softmax probabilities for $\mathcal{D}^R$ and $\mathcal{D}^U$.
   - $\mathbf{p}^R(x) \leftarrow M_{\Theta_d}(x)$ for $x \in \mathcal{D}^R$;
   - $CW^R(x) \leftarrow \mathbf{I}(\tilde{y} = y)$ for $(x, y) \in \mathcal{D}^R$;
   - $\mathbf{p}^U(x) \leftarrow M_{\Theta_d}(x)$ for $x \in \mathcal{D}^U$;
2. Train monitor models with $\mathbf{p}^R$ and CW^R.

for b = *1 to B* **do**
- Initialize $\Theta_a^{(b)}$ for a monitor model $M_{\Theta_a^{(b)}}$;
- Train $M_{\Theta_a^{(b)}}$ with $(\mathbf{p}^R(x), CW^R(x))$;
- $s^{(b)}(\mathbf{p}^U(x)) \leftarrow M_{\Theta_a^{(b)}}(\mathbf{p}(x))$ for $x \in \mathcal{D}^U$;

end

3. Actively label dataset $\mathcal{D}_s^U$ from user's dataset $\mathcal{D}^U$
   - Calculate Shannon entropy $E(x)$ based on $s^{(b)}(\mathbf{p}^U(x))$ and average over B monitor models for $(x, y) \in \mathcal{D}^U$;
   - $\mathcal{D}_s^U \leftarrow \{(x, y) \in \mathcal{D}^U | E(x)$ among the top $t\%\}$;
   - $\mathbf{p}_s^U(x) \leftarrow M_{\Theta_d}(x)$ for $(x, y) \in \mathcal{D}_s^U$;
   - $CW_s^U(x) \leftarrow \mathbf{I}(\tilde{y} = y)$ for $(x, y) \in \mathcal{D}_s^U$;
4. Transfer learning and accuracy estimation.

for b = *1 to B* **do**
- Transfer $M_{\Theta_a^{(b)}}$ with $(\mathbf{p}_s^U(x), CW_s^U(x))$;
- $s^{(b)}(\mathbf{p}^U(x)) \leftarrow M_{\Theta_a^{(b)}}(\mathbf{p}^U(x))$ for $x \in \mathcal{D}^U \setminus \mathcal{D}_s^U$;

end

return Average $\widetilde{Acc}$ from Eqn. (2);