Post-hoc Out-of-Distribution Detection

CS 726: Course Project

Harshit Varma[1], Aaron Ninan[2], Eeshaan Jain[2], Ipsit Mantri[2]

Indian Institute of Technology Bombay

[1]Department of Computer Science and Engineering
[2]Department of Electrical Engineering

OOD Detection

- Detecting 'Out-Of-Distribution' samples
- Usually aims to learn/define a scoring function that assigns high scores to ID data and low scores to OOD data
- We focus only on classification problems, after a classifier has already been trained in a standard way (a *post-hoc* setting)
- A commonly used baseline proposed by [HG16]

$$\text{softmax\_score}(x) = \max_{y \in \mathcal{Y}} p(y|x) = \frac{\max_{y \in \mathcal{Y}} \left(\exp \left(\langle y, F(x; \theta^F) \rangle \right) \right)}{\sum_{y' \in \mathcal{Y}} \exp \left(\langle y', F(x; \theta^F) \rangle \right)}$$

- Similarly can define:

$$\text{max\_logit\_score}(x) = \max_{y \in \mathcal{Y}} \left(\langle y, F(x; \theta^F) \rangle \right)$$

$$\text{avg\_logit\_score}(x) = -\frac{1}{K} \sum_{y \in \mathcal{Y}} \langle y, F(x; \theta^F) \rangle$$

Energy-based OOD Detection

- Softmax Classifiers

$$p(y|x) = \frac{\exp\left(\langle y, F(x;\theta^F)\rangle\right)}{\sum_{y' \in \mathcal{Y}} \exp\left(\langle y', F(x;\theta^F)\rangle\right)}$$

- Energy-based Models

$$p(y|x) = \frac{\exp\left(-E(x,y;\theta^E)/T\right)}{\sum_{y' \in \mathcal{Y}} \exp\left(-E(x,y';\theta^E)/T\right)} = \frac{\exp\left(-E(x,y;\theta^E)/T\right)}{\exp\left(-E(x;\theta^E)/T\right)}$$

$$E(x;\theta^E) = -T \log\left(\sum_{y \in \mathcal{Y}} \exp\left(-E(x,y;\theta^E)/T\right)\right)$$

Energy-based OOD Detection

- Can view a classifier as an energy-based model

$$E(x, y; \theta^E) = E(x, y; \theta^F) = -T\langle y, F(x; \theta^F)\rangle$$

$$E(x; \theta^E) = E(x; \theta^F) = -T \log\left(\sum_{y \in \mathcal{Y}} \exp\left(\langle y, F(x; \theta^F)\rangle\right)\right)$$

- Use $E(x; \theta^F)$ to score [LWOL20]

$$\text{energy\_score}(x) = -E(x; \theta^F) = \log\left(\sum_{y \in \mathcal{Y}} \exp\left(\langle y, F(x; \theta^F)\rangle\right)\right)$$

Relation between energy_score and softmax_score

- energy_score and softmax_score are related as follows

$$\log \text{softmax\_score}(x) = \log \max_{y \in \mathcal{Y}} p(y|x)$$

$$= \log \max_{y \in \mathcal{Y}} \exp\left(\langle y, F(x; \theta^F) \rangle\right) - \log \left(\sum_{y' \in \mathcal{Y}} \exp\left(\langle y', F(x; \theta^F) \rangle\right) \right)$$

$$\log \text{softmax\_score}(x) = \text{max\_logit\_score}(x) - \text{energy\_score}(x)$$

- softmax_score unreliable, as composed of two different scores acting in opposite directions

Asymptotic behaviour of energy_score

Let $l_k = \langle y_k, F(x;\theta^F)\rangle$ (the k^{th} logit)
Let $M = \text{max\_logit\_score}(x) = \max_{y\in\mathcal{Y}}\left(\langle y, F(x;\theta^F)\rangle\right)$, let this be achieved at the m^{th} logit.

$$\begin{aligned}
\text{energy_score}(x) &= \log\left(\sum_{y\in\mathcal{Y}} \exp\left(\langle y, F(x;\theta^F)\rangle\right)\right)\\
&= \log\left(\sum_{k=1}^{K} \exp\left(l_k\right)\right)\\
&= \log\left(\exp\left(M\right)\cdot\sum_{k=1}^{K} \exp\left(l_k - M\right)\right)\\
&= M + \log\left(1 + \sum_{k\neq m} \exp\left(l_k - M\right)\right)
\end{aligned}$$

Second term $\to 0$ for a 'good' classifier on ID data $\implies$ $\text{energy\_score}(x) \approx \text{max\_logit\_score}(x)$
Also observed in practice.

Dirichlet-based OOD Detection

- Assume a Dirichlet distribution over the softmax-ed logits of the DNN
- Estimate concentration parameters α via maximum likelihood

$$D = \{s^{(i)} = \mathrm{softmax}(F(x^{(i)}; \hat{\theta}^F))\}_{i=1}^{N}$$

$$\begin{aligned}\mathrm{NLL}(\alpha) &= \sum_{i=1}^{N} \left(\sum_k \log \Gamma(\alpha_k) - \log \Gamma \left(\sum_k \alpha_k \right) - \sum_k \left((\alpha_k - 1) \log s_k^{(i)} \right) \right) \\ &= N \sum_k \log \Gamma(\alpha_k) - N \log \Gamma \left(\sum_k \alpha_k \right) - \sum_k \left((\alpha_k - 1) \sum_i \log s_k^{(i)} \right)\end{aligned}$$

- Get $\hat{\alpha} = \mathrm{argmin}_{\alpha > 0} \mathrm{NLL}(\alpha)$ via gradient descent. Adam converges after a few epochs.
- Define dirichlet_score as follows

$$\mathrm{dirichlet\_score}(x) = -\sum_k \left((\hat{\alpha}_k - 1) \sum_i \log s_k^{(i)} \right)$$

Asymptotic behaviour of dirichlet_score

- For a good classifier $F(x; \hat{\theta}^F)$, expected to have $\alpha_k \approx \alpha_0 \; \forall \; k \in \{1, \ldots, K\}$ with $\alpha_0 \ll 1$
- Corresponds to a Dirichlet distribution having the density concentrated at the corners of the simplex $\mathcal{S}_{K-1}$
- Check behaviour of $\log p(s|\alpha)$ when $\alpha_k = \alpha_0 \; \forall \; k \; \alpha_0 \to 0^+$ (see report for full derivation)

$$\lim_{\alpha_0 \to 0^+} \log p(s|\alpha) = \lim_{\alpha_0 \to 0^+} \left(\log \Gamma(K\alpha_0) - \sum_k \log \Gamma(\alpha_0) \right) - \sum_k \log s_k$$
$$\propto K \left(\text{energy\_score}(x) + \text{avg\_logit\_score}(x) \right)$$

- dirichlet_score acts as an ensemble of two different score functions
- Can be reason behind the consistent improvements observed over the energy_score

Finetuning with dirichlet_score

- The NLL loss defined earlier leads to a natural auxiliary loss function which can be used to fine-tune the model when auxiliary OOD data is available
- α's fixed to the values obtained after fitting to the ID data
- We aim to calibrate the softmax probabilities of the ID data towards the learnt probability distribution and the OOD data anywhere away from it
- X_{in}, X_{out} are batches of ID and OOD data respectively. $t_k^{(j)}$ is the softmax probability of the k^{th} class for the j^{th} sample in the OOD batch. $s_k^{(i)}$ defined in a similar way for X_{in}.

$$
\begin{aligned}
L_{ft}(X_{in}, X_{out}) &= \sum_k \left((\alpha_k - 1) \sum_i \log t_k^{(i)} \right) - \sum_k \left((\alpha_k - 1) \sum_i \log s_k^{(i)} \right) \\
&= \sum_k (\alpha_k - 1) \left(\sum_j \log t_k^{(j)} - \sum_i \log s_k^{(i)} \right)
\end{aligned}
$$

- The below loss can then be used for fine-tuning

$$
L(X_{in}, Y_{in}, X_{out}) = L_{ce}(X_{in}, Y_{in}) + \lambda L_{ft}(X_{in}, X_{out})
$$

Finetuning with energy_score

- Similar to the previous section, energy_score can be used for finetuning the neural network so that in-distribution-based energies are assigned a lower value and out-of-distribution data is assigned higher values
- This allows for more distinguishable in-/out-of-distribution data as we have more flexibility in shaping the energy surface
- The paper suggested a Dual Margin Loss (DML) which can be appended to the cross-entropy loss in a similar fashion as Dirichlet, with the expression

$$
\begin{aligned}
L_{\text{energy}} = & \mathbb{E}_{(\mathbf{x}_{\text{in}}, y) \sim \mathcal{D}_{\text{in}}^{\text{train}}} \left(\max(0, E(\mathbf{x}_{in}) - m_{\text{in}}) \right)^2 \\
& + \mathbb{E}_{(\mathbf{x}_{\text{out}}, y) \sim \mathcal{D}_{\text{out}}^{\text{train}}} \left(\max(0, m_{\text{out}} - E(\mathbf{x}_{out})) \right)^2
\end{aligned}
$$

- To set m_{in}, first we find $\mathbb{E}(E(\mathbf{x}_{in}))$ and set it to a value lower than that. For m_{out}, we find $\mathbb{E}(E(\mathbf{x}_{out}))$ where the data is auxiliary, and set m_{out} to be larger than the obtained value
- Tuning the two margin hyperparameters requires careful tuning, and we claim that having two margins are unnecessary for the task

Analysis of L_{energy} (1)

- The goal of finetuning and the corresponding loss is to lower the energies of the in-distribution data and increase of the out-of-distribution data
- We need to heavily penalize those out-of-distribution energies which lie near in-distribution energy ranges. With this intuition, we describe three loss functions which we tested upon, with the motivation in brackets
- MCL (Minimum Classification Error)

$$L_{\text{energy}} = \mathbb{E}_{\substack{(\mathbf{x}_{\text{in}}, y) \sim \mathcal{D}_{\text{in}}^{\text{train}} \\ (\mathbf{x}_{\text{out}}, y) \sim \mathcal{D}_{\text{out}}^{\text{train}}}} \left[\frac{1}{1 + e^{-(E(\mathbf{x}_{in}) - E(\mathbf{x}_{out}))}} \right]$$

Analysis of L_{energy} (1)

- LOL(Log/Hinge)

$$L_{\text{energy}} = \mathbb{E}_{\substack{(\mathbf{x}_{\text{in}}, y) \sim \mathcal{D}_{\text{in}}^{\text{train}} \\ (\mathbf{x}_{\text{out}}, y) \sim \mathcal{D}_{\text{out}}^{\text{train}}}} \left[\log \left(1 + e^{E(\mathbf{x}_{in}) - E(\mathbf{x}_{out})} \right) \right]$$

- HEL (Harmonic Energy)

$$L_{\text{energy}} = \mathbb{E}_{\substack{(\mathbf{x}_{\text{in}}, y) \sim \mathcal{D}_{\text{in}}^{\text{train}} \\ (\mathbf{x}_{\text{out}}, y) \sim \mathcal{D}_{\text{out}}^{\text{train}}}} \left[- \frac{2E(\mathbf{x}_{out})}{1 + E(\mathbf{x}_{in}) \cdot E(\mathbf{x}_{out})} \right]$$

- All are parameterless loss functions! Empirically these loss functions beat DML

Evaluation

- Datasets: MNIST, FMNIST, CIFAR-10, MNIST-35689 (i.e., only the classes 3, 5, 6, 8 and 9 of MNIST)
- Model: VGG-16
- Metrics
  - FPR95: FPR of OOD samples when the TPR for ID samples is 95%. Classification threshold set at the 95^{th} percentile of the ID scores.
  - AUROC: The area under the receiver operating characteristic
  - AUPR: Area under the Precision-Recall curve
- Finetuning settings
  - No auxiliary dataset available: random patching used to create synthetic auxiliary data from the ID data
  - Auxiliary dataset available: a completely different dataset is used for finetuning

Results without any finetuning

ID Dataset	OOD Dataset	FPR95 (S)	FPR95 (E)	FPR95 (D)	AUROC (S)	AUROC (E)	AUROC (D)	AUPR (S)	AUPR (E)	AUPR (D)
MNIST	CIFAR10	0.0093	0.0105	**0.0080**	0.9927	0.9948	**0.9953**	0.9945	0.9956	**0.9962**
MNIST	FMNIST	0.0332	0.0341	**0.0250**	0.9884	0.9910	**0.9921**	0.9911	**0.9925**	0.9921
FMNIST	CIFAR10	0.6675	0.3916	**0.3645**	0.8790	0.9243	**0.9331**	0.9015	0.9309	**0.9400**
FMNIST	MNIST	0.7589	0.5543	**0.5361**	0.8105	0.8578	**0.8706**	0.8391	0.8700	**0.8829**
CIFAR10	MNIST	0.6261	0.4661	**0.3996**	0.8657	0.9128	**0.9263**	0.8897	0.9278	**0.9380**
CIFAR10	FMNIST	0.6038	0.4379	**0.3552**	0.8815	0.9232	**0.9393**	0.9056	0.9373	**0.9496**
MNIST_35869	MNIST_01247	**0.4117**	0.4437	0.4260	0.9282	0.9224	**0.9288**	0.9358	0.9310	**0.9360**
MNIST_35869	CIFAR10	0.0824	0.0937	**0.0555**	0.9776	0.9809	**0.9849**	0.9752	0.9762	**0.9811**

Table: S: softmax_score, E: energy_score, D: dirichlet_score

- All scores perform very well on MNIST
- Rest are the interesting cases, especially MNIST_35689 vs MNIST_01247 as the softmax_score performs better than both the scores in this case

Results after finetuning with Dirichlet Loss (No aux. setting)

ID Dataset	OOD Dataset	F1-score (ID, D, D)	F1-score (ID, E, DM)	FPR95 (E, DM)	FPR95 (D, D)	AUROC (E, DM)	AUROC (D, D)	AUPR (E, DM)	AUPR (D, D)
FMNIST	CIFAR10	0.9195	0.9251	**0.1909**	0.2539	**0.9716**	0.9513	**0.9751**	0.9558
FMNIST	MNIST	0.9195	0.9251	**0.2081**	0.4337	**0.9672**	0.9060	**0.9708**	0.9129
MNIST_35869	MNIST_01247	0.9885	0.9940	**0.2337**	0.3704	**0.9555**	0.9102	**0.9574**	0.9127
CIFAR10	MNIST	0.8763	0.8699	0.3914	**0.2473**	0.9317	**0.9566**	0.9426	**0.9645**
CIFAR10	FMNIST	0.8763	0.8699	0.3932	**0.2166**	0.9326	**0.9640**	0.9428	**0.9701**

Table: (E, DM): energy_score after finetuning with the Dual Margin Loss, (D, D): dirichlet_score after finetuning with the dirichlet loss

- Finetuning with both the losses improve the metrics, compared to the corresponding cases of no finetuning
- DML has 2 hyperparameters while Dirichlet loss has none

Results after finetuning with Dirichlet Loss (No aux. setting): Plots

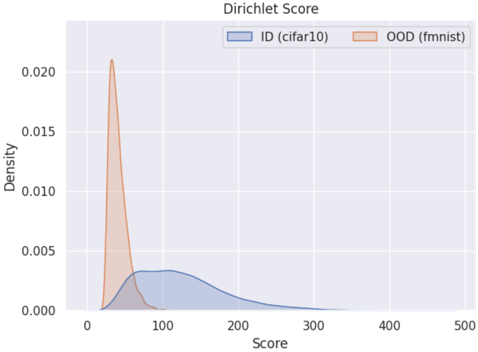

Figure: Left: Before finetuning with Dirichlet loss, Right: After finetuning with Dirichlet loss

Results after finetuning with Energy losses (Aux. setting)

Loss	F1-score (ID)	FPR95 (E)	FPR95 (D)	AUROC (E)	AUROC (D)	AUPR (E)	AUPR (D)
DML	0.9834	0.5381	0.5207	0.8131	0.7951	0.7820	0.7642
MCL	0.9944	0.2443	0.2626	0.9458	0.9364	0.9438	0.9323
LOL	**0.9954**	**0.2067**	**0.2172**	**0.9641**	**0.9591**	**0.9668**	**0.9617**
HEL	0.9927	0.3265	0.3176	0.9387	0.9353	0.9448	0.9388

Table: ID dataset: MNIST_35689, OOD dataset: MNIST_01247, Finetune dataset: CIFAR10

Loss	F1-score (ID)	FPR95 (E)	FPR95 (D)	AUROC (E)	AUROC (D)	AUPR (E)	AUPR (D)
DML	0.9908	0.0597	0.0960	0.9872	0.9814	0.9882	0.9824
MCL	**0.9924**	**0.0102**	**0.0120**	0.9938	0.9937	0.9950	0.9948
LOL	0.9897	0.0256	0.0243	0.9919	0.9923	0.9933	0.9936
HEL	0.9919	0.0147	0.0139	**0.9948**	**0.9947**	**0.9956**	**0.9955**

Table: ID dataset: MNIST, OOD dataset: FMNIST, Finetune dataset: CIFAR10

Results after finetuning with Energy losses (Aux. setting): Plots

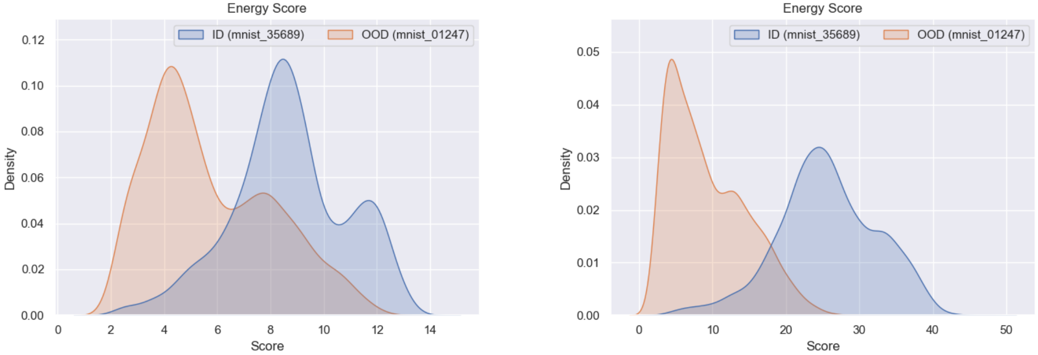

Figure: Left: Distribution plot for DML, Right: Distribution plot for LOL

- Better separation of energy values, less mixing

Conclusion + Contribution

- Presented asymptotic analysis of various scores, their inter-relatedness, and a novel score based on the Dirichlet distribution that outperforms the energy_score consistently across different metrics and datasets
- Finetuning (in both settings) improves performance by increasing the ID-OOD energy gap
- Having more parameters/margins doesn't improve performance (moreover gives worse in many cases). We can avoid extra tuning of hyperparameters by relying on any of the above margin-less losses
- Contribution:
  - Everyone: Literature survey, running models, debugging, writing report & presentation
  - Harshit: Dirichlet-based OOD formulation and analysis, finetuning in no aux. data settings
  - Eeshaan: Margin-less loss formulation and analysis, finetuning in aux. data settings
  - Aaron: Attempts at Wasserstein-distance-based score and analysis
  - Ipsit: Attempts at adversarial robustness and analysis

References

- Dan Hendrycks and Kevin Gimpel, *A baseline for detecting misclassified and out-of-distribution examples in neural networks*, arXiv preprint arXiv:1610.02136 (2016).
- Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li, *Energy-based out-of-distribution detection*, Advances in Neural Information Processing Systems **33** (2020), 21464–21475.