Supplementary Material: Bayesian Evidential Deep Learning for Online Action Detection

Hongji Guo, Hanjing Wang, and Qiang Ji

Rensselaer Polytechnic Institute, Troy NY 12180, USA
{guoh11, wangh36, jiq}@rpi.edu

This document contains the supplementary materials of Bayesian Evidential Deep Learning (BEDL). We first include the derivation of loss function (§ 1) and uncertainties (§ 2) in the main paper. Then we provide the feature extraction details including RGB features and optical flow features (§ 3). Next, we show the detailed model architectures of teacher model and student model (§ 4). The full distribution distillation algorithm is summarized in § 5. In the end, we present the details of online anomaly detection (§ 6).

1 Derivation of Loss Function

$$
\begin{aligned}
\mathcal{L}_{dis} &= KL(p(\lambda|x,\mathcal{D})||p(\lambda|\alpha(x,\psi))) \\
&\propto -\int p(\lambda|x,\mathcal{D})\log p(\lambda|\alpha(x,\psi))d\lambda \\
&= -\int\int p(\lambda|x,\theta)p(\theta|\mathcal{D})[\log p(\lambda|\alpha(x,\psi))]d\lambda d\theta \\
&= -\int p(\theta|D)[\log p(\lambda(x,\theta)|\alpha(x,\psi))]d\theta \\
&= -\sum_{c=1}^{C}\log(\Gamma(\alpha_c)) + \log\Gamma(\sum_{c=1}^{C}\alpha_c) - \mathbb{E}_{p(\theta|\mathcal{D})}[\sum_{c=1}^{C}(\alpha_c-1)\log\lambda_c(x,\theta)]
\end{aligned}
\tag{1}
$$

2 Derivations of Uncertainties

Here we provide the derivation of total uncertainty and epistemic uncertainty in Eq. (9) of the main paper.

$$
\begin{aligned}
\mathcal{H}[p(y|x,\theta)] &= -\sum_{c=1}^{C}p(y_c|x,\mathcal{D})\log p(y_c|x,\mathcal{D}) \\
&= \sum_{c=1}^{C}\frac{\alpha_c}{\alpha_0}\log\frac{\alpha_c}{\alpha_0},\ \text{where } \alpha_0 = \sum_{c=1}^{C}\alpha_c
\end{aligned}
\tag{2}
$$

$$
\begin{aligned}
\mathcal{I}[y;\lambda|\alpha] &= \mathcal{H}[p(y|x,\theta)] - E_{p(\lambda|x,\theta)}[\mathcal{H}[p(y|\lambda)]] \\
&= -\sum_{c=1}^{C}\frac{\alpha_c}{\alpha_0}\big(\ln\frac{\alpha_c}{\alpha_0}\Psi(\alpha_c+1) + \Psi(\alpha_0+1)\big)
\end{aligned}
\tag{3}
$$

3 Feature Extraction

We use TSN [3] to extract RGB features, which can be accessed at https://github.com/yjxiong/anet2016-cuhk. Video frames are extracted at 24 fps and the chunk size is set to 6. The model is pretrained on ActivityNet 1.3 and Kinetics-400 for action recognition. Specifically, the output of the RGB features are in 2048-dimension for ActivityNet and in 3072-dimension for Kinetics. On the other hand, the optical flow features are extracted by BN-Inception [1]. Firstly, we extract the optical flows by DenseFlow. Specifically, the output flow features are in 1024-dimension. We concatenate the RGB features and optical flow features at each frame as the final input, which leads to 3072 dimensions for ActivityNet pretrained features and 4096 dimensions for Kinetics pretrained features.

4 Detailed Model Architectures

Bayesian teacher model. The teacher model follows a multi-head transformer architecture. Given the input feature vectors, we first add the positional embedding to retain the positional information. We follow the same procedures in the original transformer [2]. The positional embedding can be computed as:

$$\begin{aligned} PE_{(pos,2i)} &= sin(pos/10000^{2i/C}) \\ PE_{(pos,2i+1)} &= cos(pos/10000^{2i/C}) \end{aligned} \tag{4}$$

where pos is the input position and i is the dimension index. Then the input is linear projected to queries (Q), keys (K), and values (V) for the following self-attention operations. The self-attention can be calculated as:

$$Att(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{5}$$

where d_k is the dimension of query, key, and value.

Denote the input as X, the multi-head attention procedures can be summarized as below:

$$\begin{aligned} Z_1 &= MultiHeadAtt(Norm(X)) \\ Z_2 &= Z_1 + X \\ Z_{out} &= FFN(Norm(Z_2)) + Z_2 \end{aligned} \tag{6}$$

where $Norm$ denotes layer normalization and FFN denotes feed-forward network. Z_{out} is the final output of the encoder layer. In this work, we only adopt the transformer encoder to capture the global dependency of historical frames. Specifically, we set the number of head as 6 and stack 5 encoder layers for the teacher model.

After the multi-head attention, the output embedding of the last token corresponding to the current frames are fed into a two-layer fully-connected MLP for final action classification.

Evidential teacher model. The student model is also based on the multi-head transformer. Different from the teacher model, we reduce the number of layers (2), heads (2), and hidden dimensions (256) to improve the inference efficiency.

5 Full Distribution Distillation Algorithm

Here we summarize the full distribution distillation procedures of BEDL in Algorithm 1, which including both teacher model training and student model training.

Algorithm 1 Distribution distillation procedures

Input: $\mathcal{D} = \{(x_n, y_n)\}_{n=1}^{N}$ - Training data, where x_n is the data sample, y_n is the label, and N is the total number of samples in the training set.
Output: θ_s - parameters of student model

1 - Training Bayesian teacher model

1.1 - Training of deterministi teacher model

Denote the parameters of the teacher model as $\Theta = \{\phi, \theta\}$, where ϕ includes the parameters before the last layer and θ includes the parameters of last layer.
for $n = 1$ to N **do**
  Make the prediction of x_n deterministically: $p(\hat{y}_n|x_n, \phi, \theta) \in \mathbb{R}^C$, where C is the total number of action classes
  Compute the teacher cross-entropy loss $\mathcal{L}_{CE}^t = -\sum_{c=1}^{C} \mathbb{1}(y_n = c) \log p(\hat{y}_n = c|x_n, \phi, \theta) + r(\phi, \theta)$, where $r(\phi, \theta)$ is a regularizer (a.k.a. weight decay)
  Optimizing ϕ_t and θ by minimizing $\mathcal{L}_{CE}^t$
end for
Save the optimized model parameters ϕ^* and θ^*

1.2 - Laplace Approximation (last-layer)

Using LA technique to obtain $p(\theta|\mathcal{D}) \sim \mathcal{N}(\theta^*, -H^{-1})$,
where θ^* is the point-estimate obtained from the last step and $H = \nabla_{\theta_t}^2 \log p(\theta|\mathcal{D})|_{\theta=\theta^*}$ is the Hessian matrix

1.3 - Computing mutual information

Compute the mutual information of each feature element following the procedures in Sec. 3.3

2 - Distilling knowledge to student model

2.1 - Training of evidential student model

Denote the parameters of the student model as ψ
Model: $x \to \alpha \to \lambda \to y$, where λ denotes the parameters of the categorical distribution $p(y|\lambda)$, and α is the parameters of distribution of $\lambda = Dir(\lambda|\alpha)$
for $n = 1$ to N **do**
  Generate $\alpha_n \in \mathbb{R}^C$ by feeding x_n into the model
  Sample $\lambda_n \in \mathbb{R}^C$ from $Dir(\lambda|\alpha_n)$
  Make the prediction of x_n as $p(\hat{y}_n|\lambda_n)$
  Compute the distillation loss for HPNN: $\mathcal{L}_{dis} = KL[p(\lambda_n|x, \mathcal{D}, \phi^*)||p(\lambda_n|\alpha(x, \psi))] = \sum_{c=1}^{C} \log(\Gamma(\alpha_n^c)) + \log\Gamma(\sum_{c=1}^{C} \alpha_n^c) - E_{p(\theta|\mathcal{D},\phi^*)}[\sum_{c=1}^{C}(\alpha_n^c - 1) \log \lambda_n^c(x, \phi^*, \theta)]$
  Optimizing ψ by minimizing $\mathcal{L}_{dis}$
end for
return Updated student model parameters ψ

6 Online Anomaly Detection

For THUMOS'14 dataset, the "ambiguous" class besides the 20 actions and background are treated as anomaly data. For the TVSeries dataset, we randomly select 5 classes as anomaly and using both their original training and testing split as the anomaly test set. During the training, the anomaly data are ignored. During the testing, if the estimated uncertainty from the student model is above the pre-defined threshold, the input is declared as the anomaly. We explored different uncertainty threshold and empirically set the threshold as 0.12 for best performance. The evaluation of different thresholds can be seen from the ROC curves in the main paper.

References

1. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning. pp. 448–456. pmlr (2015) 2
2. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017) 2
3. Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Van Gool, L.: Temporal segment networks: Towards good practices for deep action recognition. In: European conference on computer vision. pp. 20–36. Springer (2016) 2