Fusion of evidential CNN classifiers for image classification

Zheng Tong, Philippe Xu, Thierry Denœux

Université de Technologie de Compiègne
HEUDIASYC (UMR CNRS 7253)

October 16, 2021

Outline

1. Introduction
2. Evdential fusion of convolutional neural networks
3. Experiments

Problem definition

- Convolutional neural networks (CNNs) achieve remarkable success on image classification.
- Such CNNs are trained with different datasets, such as CIFAR-10 and ImageNet.
- The aim of the study:
  - Combine CNNs trained from such heterogenous datasets with Dempster-Shafer theory.
  - Allow the introduction of new datasets with different sets of classes at any stage.

Dempster-Shafer theory

Dempster-Shafer (DS) theory, also referred to as *Evidence Theory*,

1. Represent independent pieces of evidence by a mass function $m : 2^{\Omega} \rightarrow [0, 1]$ on the frame of discernment Ω, such that $\sum_{A \subseteq \Omega} m(A) = 1$.
2. Aggregate two mass functions m_1 and m_2 using Dempster's rule as

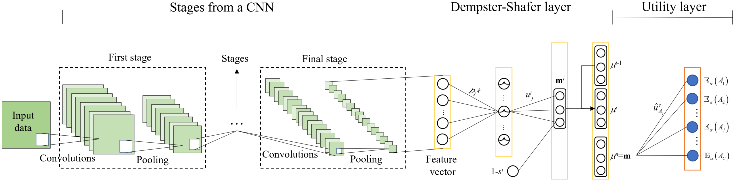
$$(m_1 \oplus m_2)(A) = \frac{\sum_{B \cap C = A} m_1(B) m_2(C)}{\sum_{B \cup C \neq \emptyset} m_1(B) m_2(C)}$$

3. Refine a frame Ω to another one Θ and compute the vacuous extension as

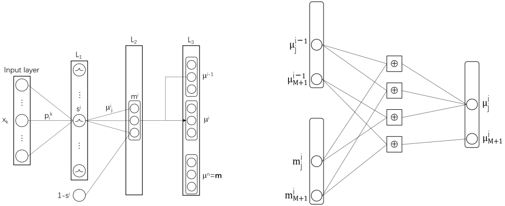
$$m^{\Omega \uparrow \Theta}(B) = \begin{cases} m^{\Omega}(A) & \text{if } \exists A \subseteq \Omega, \quad B = \bigcup_{\omega \in A} \rho(\{\omega\}), \\ 0 & \text{otherwise}, \end{cases}$$

Evidential convolutional neural network

- Plug in a "DS layer" at the backbone output of a CNN.



- A DS layer converts features into mass functions.



- Zheng Tong, Philippe Xu, and Thierry Denœux. "An evidential classifier based on Dempster-Shafer theory and deep learning". In: *Neurocomputing* 450 (2021), pp. 275–293.

Outline

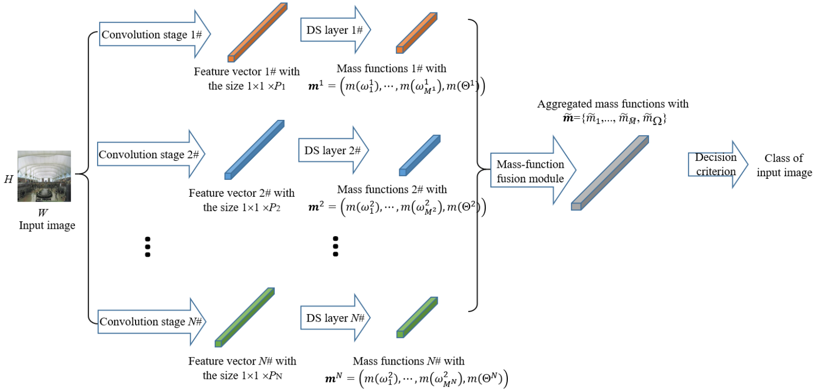
1 Introduction

2 Evdential fusion of convolutional neural networks

3 Experiments

Basic idea

- Combine different pre-trained networks for a general one.
- Adding a mass-function fusion module at the mass-function outputs of evidential CNNs.



Learning with soft labels I

- Fine-tune with union of learning sets: learned parameters in pre-trained evidential CNNs could not be suitable for the new frame.
- After merging, some label become imprecise $A_* \subseteq \Omega$, called soft label.
- Given N CNN backbones, the n-th CNN architecture with a DS layer outputs a mass function m^n on the frame of discernment Θ^n, $n = 1, \ldots, N$.
- Let Ω be a common refinement of the N frames $\Theta^1, \ldots, \Theta^N$, the vacuous extension of m^n in Ω is $m^{n\uparrow\Omega}$.
- Aggregate these vacuous extension into one on the common refinement as $\widetilde{m}$.
- Converts $\widetilde{m}$ into pignistic probability as

$$BetP_{\widetilde{m}}(\{\omega\}) = \sum_{A\subseteq\Omega;\omega\in A} \frac{\widetilde{m}}{|A|}.$$

Learning with soft labels II

- The combination of evidential CNNs outputs $\{BetP_{\widetilde{m}}(\{\omega_1\}), \ldots, BetP_{\widetilde{m}}(\{\omega_M\})\}$, and the final prediction is $\widehat{\omega} = \arg\max_{\omega \in \Omega} BetP_{\widetilde{m}}(\{\omega\})$.
- We define the loss function w.r.t the pignistic probability for a sample with label $A_* \subseteq \Omega$ as:

$$\mathcal{L}(\boldsymbol{p}, A_*) = -\log \sum_{\omega \in A_*} BetP_{\widetilde{m}}(\{\omega\}).$$

It achieves 0 when the sum of the pignistic probabilities of the classes in A_* equals to 1.

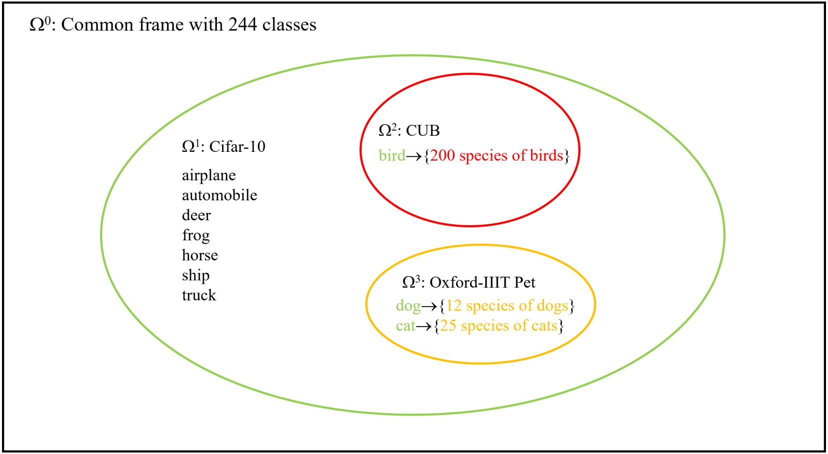
Outline

1 Introduction

2 Evdential fusion of convolutional neural networks

3 Experiments

Datasets


Ω^0: Common frame with 244 classes
Ω^1: Cifar-10
airplane
automobile
deer
frog
horse
ship
truck
Ω^2: CUB
bird→{200 species of birds}
Ω^3: Oxford-IIIT Pet
dog→{12 species of dogs}
cat→{25 species of cats}

Experiment details: CNN architectures

1. Implementation details:
   - Design a mass-fusion evidential CNN (MFE-CNN) classifiers with three pre-trained CNN backbones.
   - For each MFE-CNN classifier, its three CNN backbones refer to FitNet-4 (360,230,70).
   - All of the three CNN architectures have 128 output units.
2. Comparison study:
   - Probability-to-mass fusion (PMF) method
   - Bayesian fusion (BF) method
   - Probability-feature-concatenation (PFC) method
   - Mass-feature-concatenation (MFC) method

Results

	Classifier	CIFAR-10	CUB	Oxford-IIIT pet	Overall
Before fusion	E-FitNit-4	6.50	**25.07**	10.17	-
	P-FitNit-4	6.58	25.18	10.56	-
After fusion	MFE-FitNit-4	5.07	25.07	9.82	12.65
	PMF-FitNit-4	5.86	25.16	10.13	13.12
	BF-FitNit-4	6.10	27.84	11.08	14.31
	E2E MFE-FitNit-4	**4.49**	25.07	**9.81**	**12.37**
	E2E PMF-FitNit-4	5.47	25.14	10.11	12.92
	E2E BF-FitNit-4	6.26	27.76	10.87	14.32
	E2E PFC-FitNit-4	6.20	25.11	9.78	13.21
	E2E EFC-FitNit-4	6.86	25.10	11.30	13.80

	Classifier	aero	mobile	bird	cat	deer	dog	frog	horse	ship	truck
Before fusion	E-FitNit-4	2.4	3.9	6.4	13.5	9.0	10.1	5.6	6.8	3.5	2.7
	P-FitNit-4	1.6	2.6	8.7	15.7	9.6	12.5	4.2	5.3	1.9	2.6
After fusion	E2E MFE	2.2	3.9	1.9	6.3	8.5	3.9	5.5	6.5	3.5	2.7
	E2E PMF	1.6	2.5	5.0	12.8	9.0	9.2	4.2	5.3	1.8	2.6
	E2E BF	1.5	2.5	8.1	14.0	9.0	11.0	4.1	5.2	1.8	2.5

Class examples

Instance/label	Before fusion			MF on Ω after fusion
	MF from CIFAR-10	MF from CUB	MF from Oxford	
/bird	$m(\{\text{airplane}\}) = 0.506$ $m(\{\text{bird}\}) = 0.382$ ... $m(\Theta^1) = 0.065$	$m(\{\text{caspinan}\}) = 0.698$ $m(\{\text{horned grebe}\}) = 0.109$ ... $m(\theta_0^2) = 0.098$	$m(\{\text{samyod}\}) = 0$ $m(\{\text{pyrenees}\}) = 0.001$ ... $m(\theta_0^3) = 0.905$	$m(\{\text{airplane}\}) = 0.101$ $m(\{\text{caspinan}\}) = 0.672$ ... $m(\Omega) = 0.007$
/caspian	$m(\{\text{airplane}\}) = 0.009$ $m(\{\text{bird}\}) = 0.823$ ... $m(\Theta^1) = 0.092$	$m(\{\text{caspinan}\}) = 0.423$ $m(\{\text{horned grebe}\}) = 0.452$ ... $m(\theta_0^2) = 0.084$	$m(\{\text{samyod}\}) = 0$ $m(\{\text{pyrenees}\}) = 0.001$ ... $m(\theta_0^3) = 0.951$	$m(\{\text{caspinan}\}) = 0.415$ $m(\{\text{horned grebe}\}) = 0.450$ ... $m(\Omega) = 0.009$
/byssinian	$m(\{\text{cat}\}) = 0.742$ $m(\{\text{dog}\}) = 0.131$ ... $m(\Theta^1) = 0.032$	$m(\{\text{caspinan}\}) = 0.002$ $m(\{\text{horned grebe}\}) = 0$ ... $m(\theta_0^2) = 0.931$	$m(\{\text{byssinian}\}) = 0.412$ $m(\{\text{bengal}\}) = 0.503$ ... $m(\theta_0^3) = 0.038$	$m(\{\text{byssinian}\}) = 0.414$ $m(\{\text{bengal}\}) = 0.505$ ... $m(\Omega) = 0.005$
/keeshond	$m(\{\text{cat}\}) = 0.158$ $m(\{\text{dog}\}) = 0.705$ ... $m(\Theta^1) = 0.058$	$m(\{\text{albatross}\}) = 0.001$ $m(\{\text{horned grebe}\}) = 0$ ... $m(\theta_0^2) = 0.975$	$m(\{\text{rogdoll}\}) = 0.682$ $m(\{\text{keeshond}\}) = 0.254$ ... $m(\theta_0^3) = 0.001$	$m(\{\text{rogdoll}\}) = 0.369$ $m(\{\text{keeshold}\}) = 0.485$ ... $m(\{\text{cat}\}) = 0.021$

Conclusions and perspectives

1. Conclusions:
   - Combine different pre-trained CNNs trained from heterogeneous databases with different sets of classes.
   - Keep at least as good performance as those of the individual models on their respective databases after combination.
   - Outperform other fusion strategies.
2. Perspectives:
   - Use different CNN backbones and datasets to test the fusion method.
   - Extend the idea to semantic segmentation.