XAI-Guided Interventional Retraining for Domain Generalization

2020-03

Earlier, we talked about a method to retrain a model by changing the latent representations of inputs so as to focus on the latent features that correspond to the "core" part of the inputs. We expect the retrained model will generalize better to new domains.

There might be more direct ways to do this. We start with a dataset $\{\mathbf{x}_i, y_i\}_{i=1}^{N}$ and train a model m on it. m can be a model pretrained by others.

Then we select of subset of training examples $\{\mathbf{x}_i, y_i\}_{i=1}^{N_1}$ using X/C-perplexity. For each example $(\mathbf{x}_i, y_i)$ in the subset, we create a new example $(\mathbf{x}'_i, y_i)$, where $\mathbf{x}'_i$ is obtained from $\mathbf{x}_i$ be **removing** the pixels not covered by the heatmap for y_i. **Removing** a pixel might be:

1. Set it to 0, or
2. Set it to a random value. In this case, we can create multiple instances of $\mathbf{x}'_i$ for each $\mathbf{x}_i$, which might help.

Lastly, we fine-tune the model m using either of the two datasets

1. $\{\mathbf{x}'_i, y_i\}_{i=1}^{N_1}$, or
2. $\{\mathbf{x}_i, y_i\}_{i=1}^{N} \cup \{\mathbf{x}'_i, y_i\}_{i=1}^{N_1}$

During fine-tuning, the model needs to learn to classify the inputs based on the core parts, and hence should generalize better to new domains. Another perspective: Spurious correlations are removed in $\mathbf{x}'_i$. So, the method can be understood as a data augmentation method for remove spurious correlations.

In addition, during fine-tuning we can freeze parameters of the earlier layers, and fine-tune only those at the last l layers, where l can be 1 or a small number.

2020-02

Models trained in one domain usually do not perform well in other domains. For example, models trained on ImageNet have much lower accuracies on ImageNet V2. *Domain adaptation* is about learning models that generalize well across domains.

In this note, I describe a method for domain adaptation that consists of two training phases: (1) regular training, and (2) interventional retraining (IR). When classifying an input image, the output depends on (1) the object(s) of interest in the image, and (2) the background. The purpose of IR is to reduce the impact of the background. It is guided by XAI. So, we call it *XAI-Guided Interventional Retraining (XGIR).*

Let $\{\mathbf{x}_i, y_i\}_{i=1}^N$ be our training data. During regular training, we obtain a feature mapping $h = f_\theta(\mathbf{x})$, which converts the data into $\{h_i, y_i\}_{i=1}^N$. We also obtain a Softmax model $P(y|h, w)$. XGIR will change only w, but not θ. It does so by converting $\{h_i, y_i\}_{i=1}^N$ into $\{h'_i, y_i\}_{i=1}^N$ and then retrain a softmax model on the new data.

Given $f_\theta(\mathbf{x})$, $P(y|h, w)$ and $\{\mathbf{x}_i, y_i\}$, we obtain h'_i as follows:

- Run CWOX to get a XAI heatmap $\mathbf{e}_i$ for y_i
  - It could be the heatmap for the class y_i itself (See Figure 1 (b.1))
  - Or, the heatmap for the confusion cluster that contains y_i (See Figure 1 (c))
- $\mathbf{x}'_i = \mathbf{x}_i \odot \mathit{supp}(\mathbf{e}_i, \delta_1)$, where δ_1 is a small hyper-parameter, e.g., 0.01. After this step, some of the background is removed. Some foreground is also removed. Hope it does not matter much. Maybe $\delta_1 = \max_p \mathbf{e}_i(p) \times 0.15$, where p stands for pixel.
- $\mathbf{x}'_i$ is the **core** part of $\mathbf{x}_i$ that contains the evidence for the class y_i. It is the part the a good model should consider when classifying $\mathbf{x}_i$ into class y_i.
- Let $\mathbf{r}_i = f_\theta(\mathbf{x}'_i) - f_\theta(\mathbf{x}^0))$, where $\mathbf{x}^0$ is the empty image. $\mathbf{r}_i$ is a vector over the units of the feature layer. Ideally, the classifier should relying those units with high values $\mathbf{r}_i(u)$.
- We **intervene** to change feature values $h_i(u)$ for units with low $\mathbf{r}_i(u)$ via **randomization**:

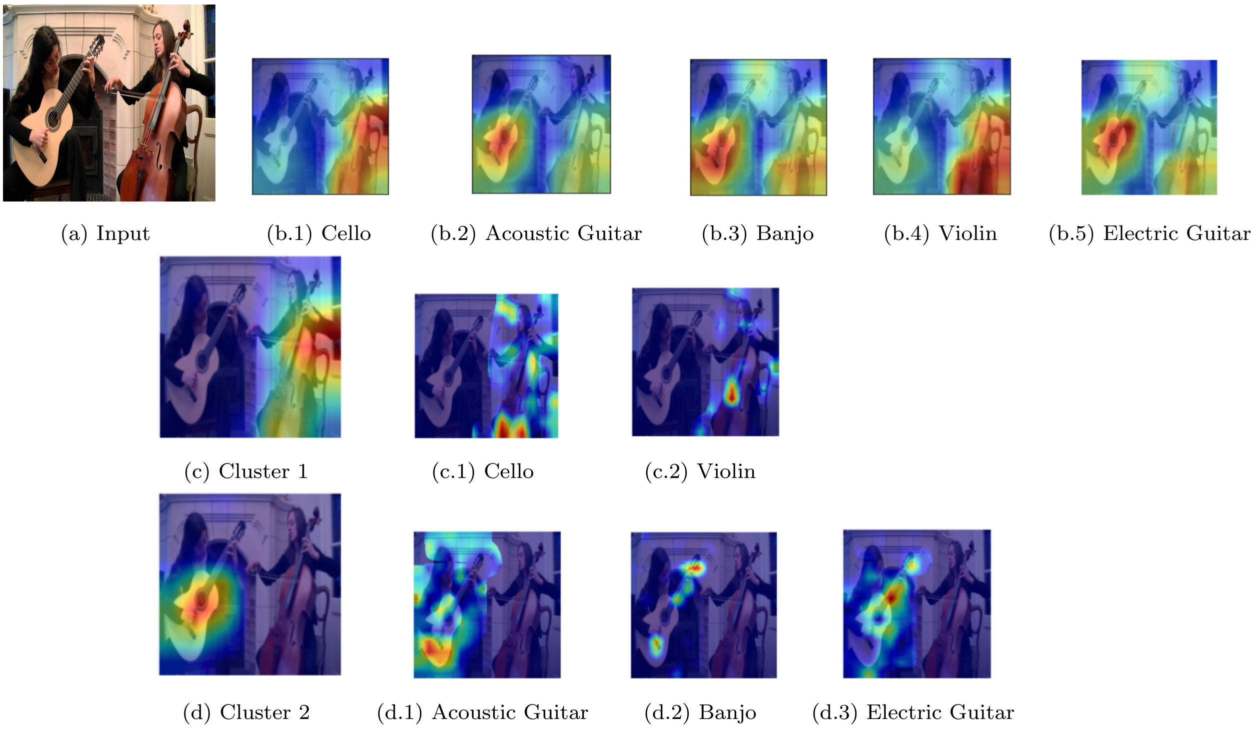
  $$h'_i(u) = (1 - \lambda_i(u))h_i(u) + \lambda_i(u)\epsilon_i,$$

  where ϵ_i is random noise generated from a Gaussian distribution, and

  $$\lambda_i(u) = \frac{1}{\delta_2}[\delta_2 - \min\{\delta_2, \mathbf{r}_i(u)\}].$$

  The hyper-parameter δ_2 determines which units are considered to be the core units that capture the information that y_i should rely on. Maybe $\delta_2 = \max_u \mathbf{r}_i(u) \times 0.5$, where p stands for pixel.

  In fact, if $\mathbf{r}_i(u) \geq \delta_2$, than $h'_i(u) = h_i(u)$. The feature is not changed as all. If $\mathbf{r}_i(u) = 0$, on the other hand, $h'_i(u) = \epsilon_i$ is a purely random value. **The model needs to learn not to rely on it**.
- Do the retraining on a subset of examples with low C-perplexity and X-Perplexity



(a) Input (b.1) Cello (b.2) Acoustic Guitar (b.3) Banjo (b.4) Violin (b.5) Electric Guitar

(c) Cluster 1 (c.1) Cello (c.2) Violin

(d) Cluster 2 (d.1) Acoustic Guitar (d.2) Banjo (d.3) Electric Guitar

Figure 1: Individual output explanation (IOX), simple whole-output explanation (SWOX), and contrastive whole-output explanation (CWOX): (a) Input image with ground-truth label `cello`; (b.1) Grad-CAM heatmap for the top class (IOX); (b.1) - (b.5) Grad-CAM heatmaps for all 5 top classes (SWOX); (c) - (d) CWOX: Contrastive heatmaps are generated to first contrast the two confusion clusters (c, d), and then to contrast the classes in each confusion cluster against each other (c.1, c.2; d.1, d.2, d.3).