Adversarial Learning

Advisor: Kris Kitani

Name: Qiqi Xiao

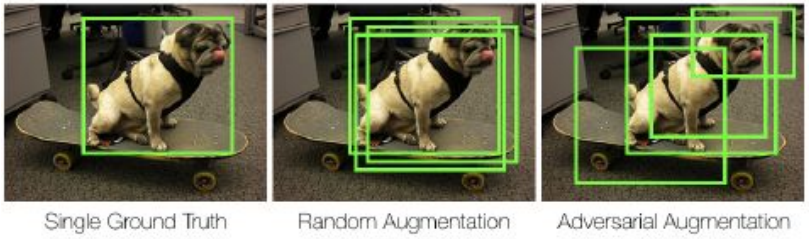
Date: 01/26/2018

ADA(Adversarial data augmentation): A Game-Theoretic Perspective on Data Augmentation for Object Detection

- Introduce an adversarial function to generate (some distribution of) maximally perturbed version of the groundtruth which is hardest for the predictor to learn.
  - Why data augmentation: ground-truth wrong/not accurate….
  - How to add data augmentation: random translation, flipping, scaling…(manually add perturbations)
  - Problems: can be error-prone

- Adversary is not free but with constraints [e.g. features(new bb) ≈ features(ori bb)].

- First work to provide theoretic basis for data augmentation in terms of an adversarial two player zero-sum game.
  - predictor(maximize performance) *vs* constrained adversary(minimize expected performance).



Problem Formulation

- Annotation distribution without augmentation:

$$p(y|x) = \begin{cases} 1, & if\ y = y^* \\ 0, & otherwise \end{cases}$$

- Annotation distribution after data augmentation:

$\tilde{p}(y|x)$: a soft distribution over labels

- Expected loss:

$$\sum_{y \in \mathcal{Y}} P(y|\boldsymbol{x}) \ell(\hat{y}, y).$$

- Probabilistic predictor:

f(y|x)

- Expected loss(Empirical Risk Minimization)

$$\min_{f \in \Gamma} \sum_{x \in D} \overbrace{\sum_{y'} f(y'|x) \sum_{y} P(y|x) \ell(y', y)}^{\text{expected loss for input } x}.$$

- Expected loss under worst case distribution

→ Adversarial Data Augmentation

$$\min_{f \in \Gamma} \sum_{x \in D} \sum_{y'} f(y'|x) \max_{P(y|x)} \sum_{y} P(y|x) \ell(y', y).$$

(Unreasonable)

Game Formulation

The value/payoff of the game for x (the expected loss)

$$\mathbb{E}_{\substack{y'|\boldsymbol{x} \sim f \\ y|\boldsymbol{x} \sim P}} [\ell(y', y)] = \sum_{y', y} f(y'|x)\ell(y', y)P(y|x)$$
$$= \mathbf{f}^\top \mathbf{G}\mathbf{p}.$$

f: the vector of probabilities obtained from the predictor over all labels

G: the game matrix where each element contains the loss between two labels

p: the annotation distribution vector

Definition

Primal Adversarial Data Augmentation(ADA-P):

$$\min_f \max_P \mathbb{E}_{\substack{\boldsymbol{x} \sim \mathcal{D}, \\ y'|\boldsymbol{x} \sim f, \\ y|\boldsymbol{x} \sim P}} [\ell(y', y)] \textit{ such that:}$$

$$\mathbb{E}_{\substack{\boldsymbol{x} \sim \mathcal{D}, \\ y|\boldsymbol{x} \sim P}} [\phi(y, \boldsymbol{x})] = \mathbb{E}_{y, \boldsymbol{x} \sim \mathcal{D}} [\phi(y, \boldsymbol{x})]$$ where

$$\mathbb{E}_{y, \boldsymbol{x} \sim \mathcal{D}} [\phi(y, \boldsymbol{x})] = \frac{1}{N} \sum_{n=1}^{N} \phi(y_n, \boldsymbol{x}_n),$$

The Dual Adversarial Data Augmentation(ADA-D):

$$\min_\theta \mathbb{E}_{\mathbf{x}, y^* \sim \mathcal{D}} \Big[\min_f \max_P \mathbb{E}_{\substack{y' \sim f, \\ y \sim P}} \Big[\ell(y', y) + \theta^\top \{\phi(y, \boldsymbol{x}) - \phi(y^*, \boldsymbol{x})\} \Big] \Big].$$

Adversarial Object Localization

Label Space: y is the 4 coordinates of a bounding box

distribution approximation → discretize the label space Y using a bb proposal algorithm

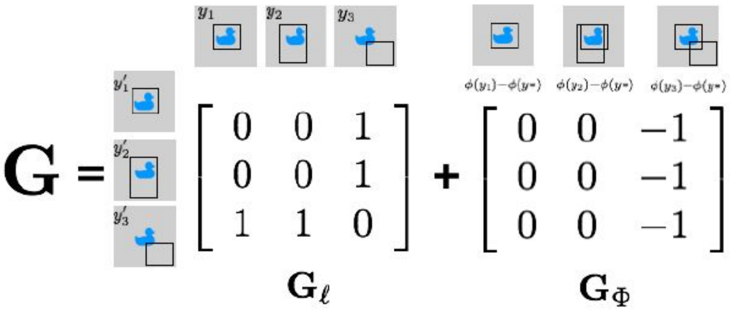
Feature statistics: $\{\phi(y', \boldsymbol{x}) - \phi(y^*, \boldsymbol{x})\}$ difference of FC7 features of the VGG16 → perceptual loss

Loss function:

$\ell(y, y') = 1 - \text{IoU}(y, y')$, or $\ell_{t\alpha}(y, y') = \begin{cases} 1 & \text{IoU}(y, y') < \alpha \\ 0 & \text{IoU}(y, y') \geq \alpha, \end{cases}$ where $IoU(y, y') = \text{area}(y \cap y')/\text{area}(y \cup y')$.

Game Matrix:

$$\mathbf{G}(y', y) = \ell(y', y) + \theta^\top\{\phi(y, \boldsymbol{x}) - \phi(y^*, \boldsymbol{x})\},$$



Nash Equilibria(solution of the game)

A pair of strategies (x, y) is said to be Nash Equilibria iff neither player can increase her expected payoff by unilaterally deviating from her strategy.

$$\max_{\mathbf{p}'} \mathbf{f}^\top \boldsymbol{G}\mathbf{p}' \leq \mathbf{f}^\top \boldsymbol{G}\mathbf{p} \leq \min_{\mathbf{f}'} \mathbf{f}'^\top \boldsymbol{G}\mathbf{p}.$$

$$\min_{v,\mathbf{f}\geq\mathbf{0}} \ v \text{ such that: } \mathbf{f}^\top \boldsymbol{G} \leq v\mathbf{1} \text{ and } \mathbf{f}^\top \mathbf{1} = 1; \ \text{and}$$

$$\max_{v,\mathbf{p}\geq\mathbf{0}} \ v \text{ such that: } \boldsymbol{G}\mathbf{p} \geq v\mathbf{1}^\top \text{ and } \mathbf{p}^\top \mathbf{1} = 1,$$

Linear programming

Constraint Generation for Large Games

To solve ADA-D without explicitly constructing the entire payoff matrix G.

Key idea: To use a set of the most violated constraints to grow a game matrix that supports the equilibrium distribution, but is much smaller than the full game matrix.

Methods: Double Oracle Algorithm

Reference:

Planning in the Presence of Cost Functions Controlled by the Adversary

Adversarial Prediction Games for Multivariate Losses

Double Oracle Algorithm

Initialization:

$\bar{R}$: all strategies the row player has played in previous iterations.

$\bar{C}$: of all the columns played by the column player.

Initialize $\bar{R}$ with an arbitrary row.

Initialize $\bar{C}$ with an arbitrary column.

Terminate conditions:

1、r_i is already in $\bar{R}$ and c_i in $\bar{C}$
2、$v_u - v_l < \varepsilon$

On iteration i

- Solve the matrix game where the row player can only play rows in $\bar{R}$ and the column player can only play columns of $\bar{C}$, using linear programming or any other convenient technique. This provides a distribution p_i over $\bar{R}$ and q_i over $\bar{C}$.
- The row player assumes the column player will always play q_i, finds an optimal pure strategy $\mathcal{R}(q_i) = r_i$ against q_i, and adds r_i to $\bar{R}$. Let $v_\ell = V(r_i, q_i)$. Since r_i is a best response we conclude that $\forall p\ V(p, q_i) \geq v_\ell$, and so we have a bound on the value of the game, $V_G = \min_p \max_q V(p, q) \geq v_\ell$.
- Similarly, the column player picks $\mathcal{C}(p_i) = c_i$, and adds c_i to $\bar{C}$. We let $v_u = V(p_i, c_i)$ and conclude $\forall q\ \ V(p_i, q) \leq v_u$, and hence $V_G = \max_q \min_p \leq v_u$.

Algorithm of ADA

Algorithm 1 ADA Equilibrium Computation

Input: Image $\boldsymbol{x}$; Parameters θ; Ground Truth y^*

Output: Nash equilibrium, $(\mathbf{f}, \mathbf{p})$

$\mathcal{Y} \leftarrow \text{EdgeBox}(\boldsymbol{x})$
$\Phi = \text{CNN}(\mathcal{Y}, \boldsymbol{x})$
$\psi \leftarrow \theta^\top(\Phi - \Phi(y^*))$
$\mathcal{S}_p \leftarrow \mathcal{S}_f \leftarrow \text{argmax}_y\, \psi(y)$
repeat
    $(\mathbf{f}, \mathbf{p}, v_p) \leftarrow \texttt{solveGame}(\psi(\mathcal{S}_p), \text{loss}(\mathcal{S}_f, \mathcal{S}_p))$
    $(y_{\text{new}}, v_{\text{max}}) \leftarrow \max_y \mathbb{E}_{y' \sim f}[\text{loss}(y, y') + \psi(y)]$
    if $(v_p \neq v_{\text{max}})$ **then**
        $\mathcal{S}_p \leftarrow \mathcal{S}_p \cup y_{\text{new}}$
    end if
    $(\mathbf{f}, \mathbf{p}, v_f) \leftarrow \texttt{solveGame}(\psi(\mathcal{S}_p), \text{loss}(\mathcal{S}_f, \mathcal{S}_p))$
    $(y'_{\text{new}}, v_{\text{min}}) \leftarrow \min_{\hat{y}} \mathbb{E}_{y \sim p}[\text{loss}(y, y')]$
    if $(v_f \neq v_{\text{min}})$ **then**
        $\mathcal{S}_f \leftarrow \mathcal{S}_f \cup y'_{\text{new}}$
    end if
until $v_p = v_{\text{max}} = v_f = v_{\text{min}}$
return $(\mathbf{f}, \mathbf{p})$

Convex optimization(gradient-based methods)

Pre-processing step, extracting box proposals and CNN features

Solve Nash equilibrium using linear programming

Experiments

Baselines: SSVM and Softmax

SSVM: Structured output SVM

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \quad \lambda||\theta||_2 + \sum_n \xi_n \tag{13}$$

$$\text{s.t.} \quad \theta^\top(\phi(y_n^*, \boldsymbol{x}_n) - \phi(y, \boldsymbol{x}_n)) \geq \ell(y_n^*, y) - \xi_n \quad \forall\, y$$

At test time

$$\hat{y} \;=\; \operatorname{argmax}_y \theta^\top \phi(y, \boldsymbol{x})$$

Softmax:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \prod_n P(y_n|\boldsymbol{x}_n;\theta), \\ = \underset{\theta}{\operatorname{argmax}} \prod_n \frac{e^{\theta^\top \phi(y_n, \boldsymbol{x}_n)}}{\sum_y e^{\theta^\top \phi(y, \boldsymbol{x}_n)}}$$

$$\hat{y} = \underset{y}{\operatorname{argmin}} \sum_{y' \in \mathcal{Y}} P(y'|\mathbf{x};\theta)\ell(y, y'),$$

Baseline comparisons with no augmentation

Table 1. No augmentation baseline comparison (IoU>0.5)

Model	ImageNet Object Categories										
	Plane	Bird	Bus	Car	Cat	Cow	Dog	Hors	Moni	Sofa	mAP
ADA+VGG (Ours)	**92.0**	**93.5**	**92.0**	**100.0**	**89.1**	**100.0**	**93.0**	**96.4**	**96.0**	**90.0**	**94.2**
Softmax+VGG	84.0	86.5	84.0	87.0	70.9	77.5	62.0	72.7	72.0	80.0	77.7
SSVM+VGG	90.0	82.5	82.0	82.0	40.0	87.5	72.0	72.7	90.0	78.0	77.7

Table 2. No augmentation baseline comparison (IoU>0.7)

Model	ImageNet Object Categories										
	Plane	Bird	Bus	Car	Cat	Cow	Dog	Hors	Moni	Sofa	mAP
ADA+VGG (Ours)	**58.0**	**61.5**	**64.0**	**91.0**	**30.9**	**77.4**	**58.0**	**58.2**	**61.8**	**61.9**	**62.3**
Softmax+VGG	47.6	45.7	40.0	62.8	20.0	42.5	25.1	25.4	31.4	44.2	38.5
SSVM+VGG	51.8	55.5	44.0	61.7	21.8	54.7	31.6	43.6	56.0	57.3	47.8

Baseline comparisons with augmentation

Table 3. Effect of Data Augmentation (IoU > 70%)

Augmentation	AlexNet Object Category										
	Plane	Bird	Bus	Car	Cat	Cow	Dog	Hors	Moni	Sofa	Avg
$SSVM_{t50}$+VGG	53.8	57.9	49.7	64.0	22.6	59.9	37.5	45.5	56.7	57.8	50.5
$SSVM_{t60}$+VGG	54.7	58.9	52.7	67.7	23.7	64.9	42.0	48.6	57.3	58.4	52.9
$SSVM_{t70}$+VGG	**56.4**	**61.6**	**56.8**	**70.8**	**25.4**	**67.3**	**49.1**	**51.9**	**58.6**	**58.8**	**55.7**
$SSVM_{t75}$+VGG	52.6	61.0	51.7	64.4	20.2	61.2	42.6	44.0	57.3	56.0	51.1
$SSVM_{t80}$+VGG	49.8	52.0	44.9	60.3	20.2	55.8	33.1	41.4	55.8	52.7	46.6
ADA+VGG (Ours)	**58.0**	**61.5**	**64.0**	**91.0**	**30.9**	**77.4**	**58.0**	**58.2**	**61.8**	**61.9**	**62.3**

Table 4. Effect of Number of Augmented Data Annotations. ADA outperforms best configuration SSVM+VGG baseline by 12%.

SSVM+VGG	k=1	k=2	k=4	k=6	k=8	k=10	k=12	ADA+VGG
mAP	77.6	79.7	81.4	**83.8**	83.7	79.8	75.3	**94.2**

Using edgebox proposal network to generate bb(s) and filter by IOU as gt(s)

Top K proposals as gt(s), use 50% IOU as success

Detection Performance Comparison

Correct label + 70% IOU

Table 5. Detection Performance Comparison (IoU > 70%).

Model	Image Net Object Category										
	Plane	Bird	Bus	Car	Cat	Cow	Dog	Hors	Moni	Sofa	Avg
ADA+VGG (Ours)	**46.0**	**55.5**	**60.0**	**86.0**	**25.4**	**70.0**	**47.0**	**52.7**	**60.0**	**48.0**	**55.1**
SSVM+VGG	42.0	46.0	38.0	53.0	16.4	52.5	25.0	36.4	42.0	42.0	39.3
Softmax+VGG	40.0	42.5	42.0	55.0	16.4	32.5	16.0	29.1	22.0	34.0	33.0

Goal and plan

- Implement from the original code and do experiments
- Apply adversarial learning data augmentation to train end-to-end detection network
- Apply it to video surveillance applications using computer graphics rendering and then maybe other types of synthetic images (like cell images)

Thank you!

Q&A