

# **Distributional smoothing with virtual adversarial training**

---

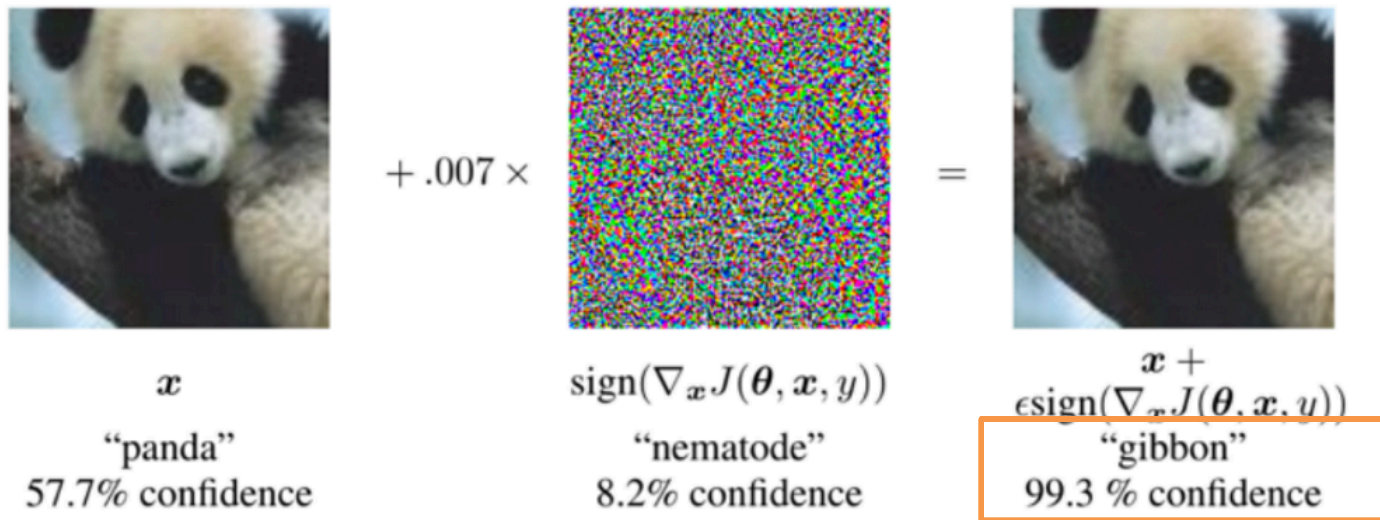
Fukuta Keisuke  
the University of Tokyo  
Harada Ushiku Lab

# Basic Information

---

- 京大のM2
- ICLR 2016 Accept (唯一の日本人?)
- 正則化としてモデルの予測分布のなめらかさを考える手法
- MNIST Semi-supervisedでSOTA (一瞬だけ)

# Adversarial example


$$\begin{array}{ccc} \text{Image of a panda} & + .007 \times \text{Adversarial perturbation} & = \text{Adversarial example (Gibbon)} \\ x & \text{sign}(\nabla_x J(\theta, x, y)) & x + \epsilon \text{sign}(\nabla_x J(\theta, x, y)) \\ \text{"panda"} & \text{"nematode"} & \text{"gibbon"} \\ 57.7\% \text{ confidence} & 8.2\% \text{ confidence} & 99.3\% \text{ confidence} \end{array}$$

gibbon

高次元になると線形識別器でも起こる問題



# Adversarial Training [Goodfellow et al. 2015]

---

- 最も間違えやすくなるようなノイズ

$$\begin{aligned} r_{adv} &= \operatorname{argmin}_r \{ p(y|x + r, \theta, |r|_p \leq \epsilon) \} \\ &= -\epsilon \overline{\nabla_x p(y|x, \theta)} \quad \left( \bar{x} \rightarrow \frac{x}{\|x\|} \right) \end{aligned}$$

- 各データサンプルに上記のノイズを加えたもの(adversarial example)も同時に学習

$$J(\theta) = \underbrace{\frac{1}{N} \sum_{n=1}^N \log p(y|x, \theta)}_{\text{元データの対数尤度}} + \lambda \underbrace{\frac{1}{N} \sum_{n=1}^N \log p(y|x + r_{adv}, \theta)}_{\text{Adversarial exampleの対数尤度}}$$


元データの対数尤度      Adversarial exampleの対数尤度

# Introduction

---

- サンプル数少なくても過学習を回避したい
  - なんらかの正則化
- モデル分布はなめらかである方がよい
  - Adversarial Example の存在は望ましくない
  - しかしAdversarial Trainingはラベルデータが必要

そこで、

 最も間違える方向ではなく、  
予測分布が最も大きく変わる方向にノイズ $r_{v-adv}$ を加える

ノイズを加えられた場合の予測分布  $p(y | x + r_{v-adv}, \theta)$  と  
元の予測分布  $p(y | x, \theta)$  が変わらないように学習

→ Unlabeled Dataも使える

→ 正則化の役割、semi-supervisedな文脈に有効

# Virtual Adversarial Training

---

ノイズを加えても分布が変わらないように学習

⇒ 2つの分布のKL-Divergenceを最小化

$$\Delta_{KL}(r, x, \theta) \equiv KL[p(y | x, \theta) || p(y | x + r, \theta)]$$

$r_{v-adv}$  はKL-Divergenceを最大化するノイズ

$$r_{v-adv} = \arg \max_r (\Delta_{KL}(r, x, \theta); |r|_2 \leq \epsilon)$$

モデルのなめらかさを図る指標としてLocal Distributional Smoothnessを導入

$$LDS(x, \theta) \equiv -\Delta_{KL}(r_{v-adv}, x, \theta)$$

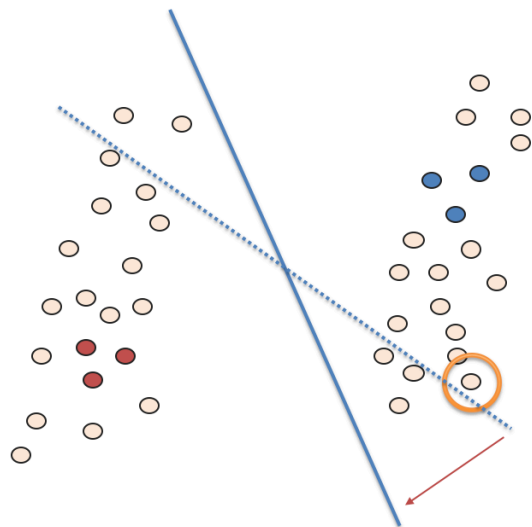
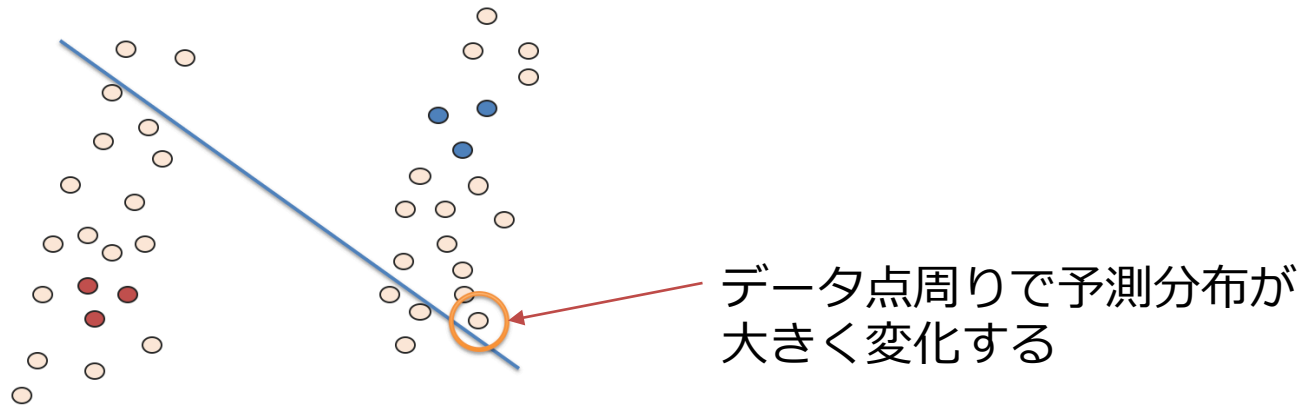
**Objective function**

$$J(\theta) = \underbrace{\frac{1}{N} \sum_{n=1}^N \log p(y|x, \theta)}_{\text{元データの対数尤度}} + \lambda \underbrace{\frac{1}{N} \sum_{n=1}^N LDS(x, \theta)}_{\text{LDSによる正則化項}}$$

元データの対数尤度

LDSによる正則化項

# イメージ



# Evaluation of $r_{v-adv}$

- Assumption
  - $p(y|x, \theta)$  can be differentiable wrt  $\theta, x$
  - $\Delta_{KL}(r, x, \theta)|_{r=0} = 0, \nabla_r \Delta_{KL}(r, x, \theta)|_{r=0} = 0$
  - $H(x, \theta) \equiv \nabla \nabla_r \Delta_{KL}(r, x, \theta)|_{r=0}$

KL情報量の $r=0$ 付近での二次までのTaylor展開

$$\Delta_{KL}(r, x, \theta) \cong \frac{1}{2} r^T H(x, \theta) r \quad \leftarrow \text{これを最大にする } r \text{ を求めたい。}$$

一般に $|x| = 1$  のとき、 $x^T A x$  の最大値は $A$  の最大固有値 $\lambda_n$ となる。  
それは $\lambda_n$ に属する固有ベクトルによって与えられる。

$$\Rightarrow r_{v-adv} = \epsilon \cdot \overline{u(x, \theta)}$$

$H(x, \theta)$ の最大固有ベクトル



# Evaluation of $r_{v-adv}$

---

$H(x, \theta)$ の最大固有ベクトルを求めたい

→べき乗法(power method)

1.  $d$  を適当に初期化
2.  $d \leftarrow \overline{H(x, \theta) \cdot d}$ を繰り返す
3.  $d$ は $H(x, \theta)$ の最大固有ベクトル  $u(x, \theta)$ に漸近する

$H$ の計算自体にも計算コスト

→有限差分法(finite difference method)

- $\frac{df}{dx} \cong \frac{f(x+h) - f(x)}{h}$  として近似すること

適当な $\xi$ を導入して $\xi d$ によって近似すると,  $H(x, \theta) \equiv \nabla \nabla_r \Delta_{KL}(r, x, \theta)|_{r=0}$ より

$$H(x, \theta) \cdot d \cong \frac{\nabla_r \Delta_{KL}(r, x, \theta)|_{r=\xi d} - \nabla_r \Delta_{KL}(r, x, \theta)|_{r=0}}{\xi d} \cdot d = \frac{\nabla_r \Delta_{KL}(r, x, \theta)|_{r=\xi d}}{\xi}$$

$$\overline{H(x, \theta) \cdot d} \cong \overline{\nabla_r \Delta_{KL}(r, x, \theta)|_{r=\xi d}}$$

# Evaluation of $r_{v-adv}$

---

結局、

1.  $d$ を適当に初期化
2.  $d \leftarrow \overline{\nabla_r \Delta_{KL}(r, x, \theta)}|_{r=\xi d}$  を  $I_p$  回繰り返す



Back propで計算できる！！

3.  $r_{v-adv} = \epsilon d$
- 何が言いたいかというと、  
KL-Divergenceを最も大きく変化させるノイズが  
一度のforward, back propで計算できる！！

# Derivative of LDS wrt $\theta$

---

## Objective function

$$J(\theta) = \underbrace{\frac{1}{N} \sum_{n=1}^N \log p(y|x, \theta)}_{\text{元データの対数尤度}} + \underbrace{\lambda \frac{1}{N} \sum_{n=1}^N LDS(x, \theta)}_{\text{LDSによる正則化項}}$$

ここまでで求めた $r_{v-adv}$ を使ってLDSをパラメータ $\theta$ で微分

$$\frac{\partial}{\partial \theta} LDS(x, \theta) = - \frac{\partial}{\partial \theta} \Delta_{KL}(r_{v-adv}, x, \theta)$$

$$\frac{\partial}{\partial \theta} LDS(x, \theta) = - \frac{\partial}{\partial \theta} KL[p(y|x, \theta') || p(y|x + r_{v-adv}, \theta)]$$

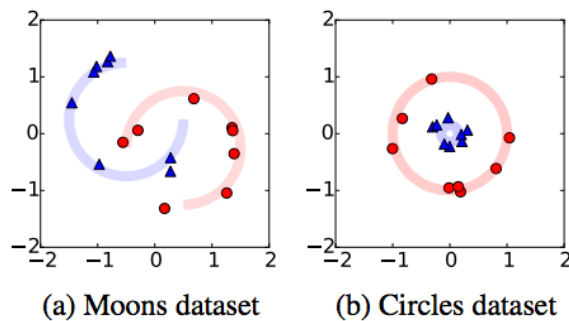
- $\nabla_{\theta} r_{v-adv}$ は無視 (そのほうがよかったらしい)
- 最初の $\theta$ も固定

# Experimental setting

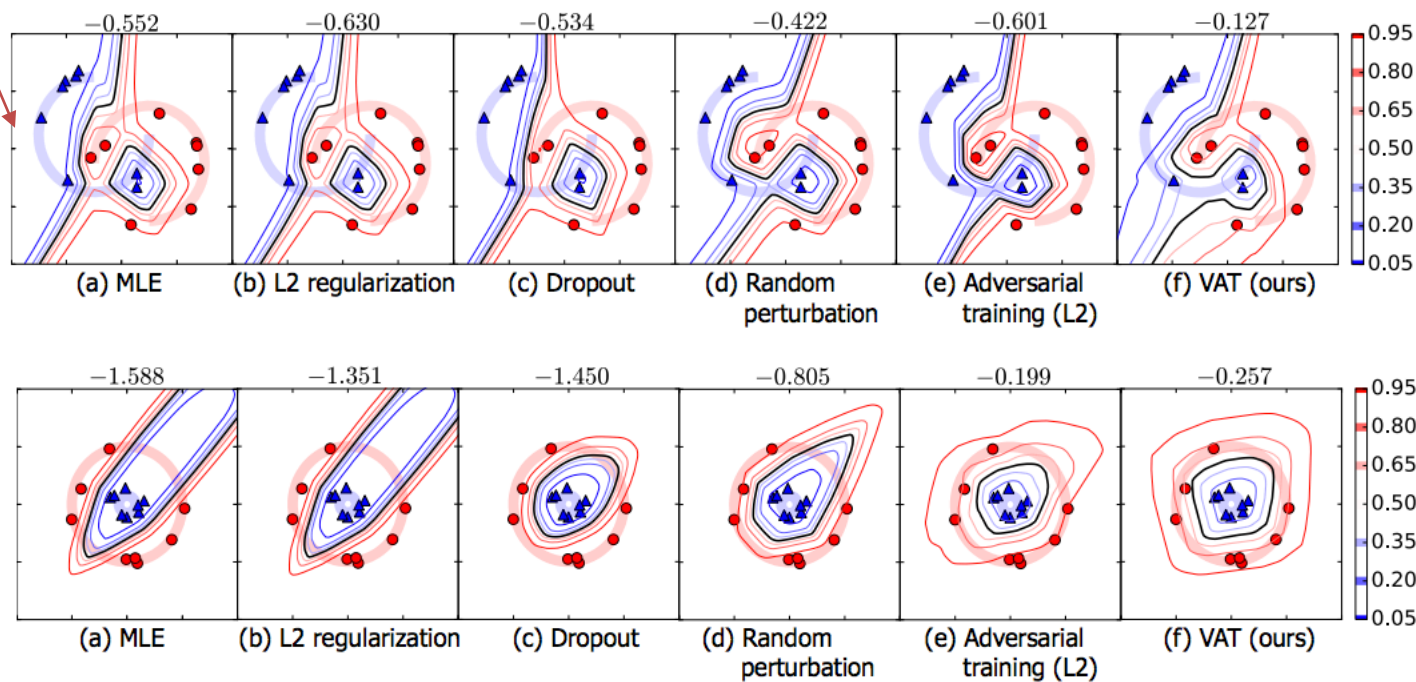
---

- $I_p = 1$ で固定,  $\lambda = 1$ で固定
- 自分で作ってみたデータセットで過学習しないかどうか
- MNISTのsupervised, semi-supervised
- Semi-supervisedにおいてラベルが無いデータについては正則化項のみを学習

# Experiments



過学習！



# Experiments

## Supervised MNIST

Method	Test error (%)
SVM (gaussian kernel)	1.40
Gaussian dropout (Srivastava et al., 2014)	0.95
Maxout Networks (Goodfellow et al., 2013)	0.94
*MTC (Rifai et al., 2011)	0.81
*DBM (Srivastava et al., 2014)	0.79
Adversarial training (Goodfellow et al., 2015)	0.782
*Ladder network (Rasmus et al., 2015)	0.57±0.02
Plain NN (MLE)	1.11
Random perturbation training	0.843
Adversarial training (with $L_\infty$ norm constraint)	0.788
Adversarial training (with $L_2$ norm constraint)	0.708
VAT (ours)	0.637±0.046

## Semi-Supervised MNIST

Method	(a) MNIST				
	$N_l$	100	600	1000	3000
SVM (Weston et al., 2012)		23.44	8.85	7.77	4.21
TSVM (Weston et al., 2012)		16.81	6.16	5.38	3.45
EmbedNN (Weston et al., 2012)		16.9	5.97	5.73	3.59
*MTC (Rifai et al., 2011)		12.0	5.13	3.64	2.57
PEA (Bachman et al., 2014)		10.79	2.44	2.23	1.91
*PEA (Bachman et al., 2014)		5.21	2.87	2.64	2.30
*DG (Kingma et al., 2014)		3.33	2.59	2.40	2.18
*Ladder network (Rasmus et al., 2015)		1.06		0.84	
Plain NN (MLE)		21.98	9.16	7.25	4.32
VAT (ours)		2.33	1.39	1.36	1.25

どちらも Ladder Network 以外には勝利!!

# Discussion

---

## 筆者のまとめ

- Random noise よりも性能良かったので  $H(x, \theta)$  利用したのは正しかった
- パラメータの取り方によらず、モデルの分布自体に制限
- 実用上使いやすい
  - ハイパラ  $\lambda$  と  $\epsilon$  だけで良い！  
 $I_p$  は1でよく、 $\xi$  もある程度小さければよい
  - $r_{v-adv}$  ,  $\frac{\partial}{\partial \theta} LDS$  にそれぞれ forward, backward 一回ずつ追加で必要なだけ
- 多様体学習取り入れたらさらに良いことがあるのでは

# VAT for semi-supervised Text Classification

[Miyato, GoodFellow+, 2016]

- Text classifierに正則化項
  - drop out より性能いいらしい
- Word-embedding空間でperturbation
  - 正規化したあとノイズ
- VATとAT組み合わせたりしてたけど基本的にVAT強い
- ちゃんとbadとgoodが遠くなる
  - 少しの変化で意味が変わらないよう学習するから

	'good'				'bad'			
	Baseline	Random	Adversarial	Virtual Adversarial	Baseline	Random	Adversarial	Virtual Adversarial
1	great	great	decent	decent	terrible	terrible	terrible	terrible
2	decent	decent	great	great	awful	awful	awful	awful
3	× <u>bad</u>	excellent	nice	nice	horrible	horrible	horrible	horrible
4	excellent	nice	fine	fine	× <u>good</u>	× <u>good</u>	poor	poor
5	Good	Good	entertaining	entertaining	Bad	poor	BAD	BAD
6	fine	× <u>bad</u>	interesting	interesting	BAD	BAD	stupid	stupid
7	nice	fine	Good	Good	poor	Bad	Bad	Bad
8	interesting	interesting	excellent	cool	stupid	stupid	laughable	laughable
9	solid	entertaining	solid	enjoyable	Horrible	Horrible	lame	lame
10	entertaining	solid	cool	excellent	horrendous	horrendous	Horrible	Horrible



# Conclusion

---

- 発想と $r_{v-adv}$ の導出手法が良い
- ImageNetまでやってほしかった。
- 発展的には
  - フィッシャー情報行列の最大固有値の最小化問題
  - フィッシャー情報行列の行列式が小さくなる
  - ガウス曲率が小さくなるので、モデルはなめらかで凹凸が少なくなる

# References

---

## 論文

- Distributional smoothing by virtual adversarial examples. [2016, Miyato et al. ]
- Explaining and Harnessing Adversarial Examples [Goodfellow et al]
- Virtual Adversarial Training for Semi-Supervised Text Classification [Miyato et al]

## 参考にした資料

- Goodfellowのスライド  
[http://www.iro.umontreal.ca/~memisevr/dlss2015/goodfellow\\_adv.pdf](http://www.iro.umontreal.ca/~memisevr/dlss2015/goodfellow_adv.pdf)
- PFNの松元さんのスライド  
<http://www.slideshare.net/eiichimatsumoto106/nips2015-ladder-network>