Data Augmentation as Feature Manipulation

Ruoqi Shen

University of Washington

Sébastien Bubeck (MSR)

Suriya Gunasekar (MSR)

Data augmentation

	no augmentation	basic augmentation	advanced augmentation
resnet18 (11M)	90%	96%	98%
cait_xxs36 (17M)	77%	88%	97%
vit_tiny (6M)	75%	86%	96%

Data augmentation as feature manipulation

Consider three types of features

1. "good" & "easy to learn"
   – accurate features with large contribution to gradients
2. "good" & "hard to learn"
   – accurate features with small contribution to gradients
3. "bad" & "easy to learn"
   – inaccurate features with large contribution to gradients

Gradient descent learns by fitting data with (1)&(3) first before using (2)

Data augmentation as feature manipulation

Consider three types of features

1. “good” & “easy to learn”
   – accurate features with large contribution to gradients
2. “good” & “hard to learn”
   – accurate features with small contribution to gradients
3. “bad” & “easy to learn”
   – inaccurate features with large contribution to gradients

Gradient descent learns by fitting data with (1)&(3) first before using (2)

Data augmentation as feature manipulation

Consider three types of features

1. “good” & “easy to learn”
   – accurate features with large contribution to gradients
2. “good” & “hard to learn”
   – accurate features with small contribution to gradients
3. “bad” & “easy to learn”
   – inaccurate features with large contribution to gradients

Gradient descent learns by fitting data with (1)&(3) first before using (2)

Data augmentation can be viewed as manipulation of relative contribution of “good” and “bad” features in the gradients, *i.e.,* make (2) -> (1), or make (3) -> “bad” & “hard to learn”

Theory: Multi-view data model Allen-Zhu & Li (2019)

- Two classes $y \in \{-1,1\}$
- Inputs x has P patches $\mathrm{x} = (\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_\mathrm{P}) \in \mathbb{R}^{d \times P}$

Theory: Multi-view data model Allen-Zhu & Li (2019)

- Two classes $y \in \{-1,1\}$
- Inputs x has P patches $\mathrm{x} = (\mathrm{x}_1, \mathrm{x}_2, \dots, \mathrm{x}_\mathrm{P}) \in \mathbb{R}^{d \times P}$
- K possible "Good features"

- "Bad features"

(noise/spurious feature)

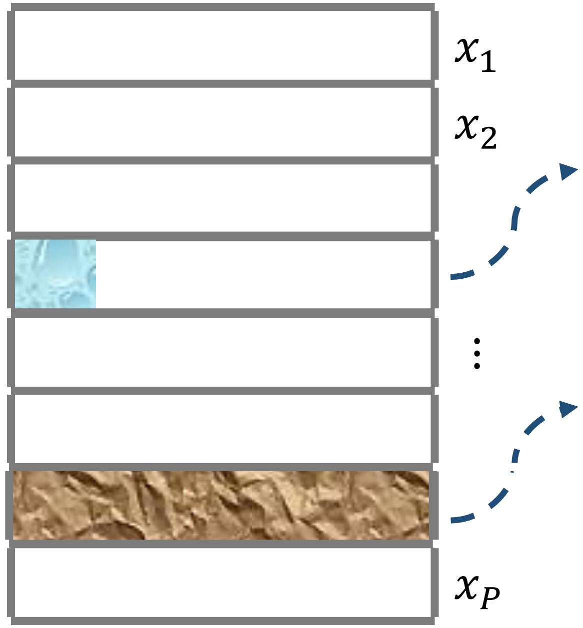
Theory: Multi-view data model

Allen-Zhu & Li (2019)

- Two classes $y \in \{-1,1\}$
- Inputs x has P patches $\mathrm{x} = (\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_\mathrm{P}) \in \mathbb{R}^{d \times P}$
- K possible "Good features"

- "Bad features"

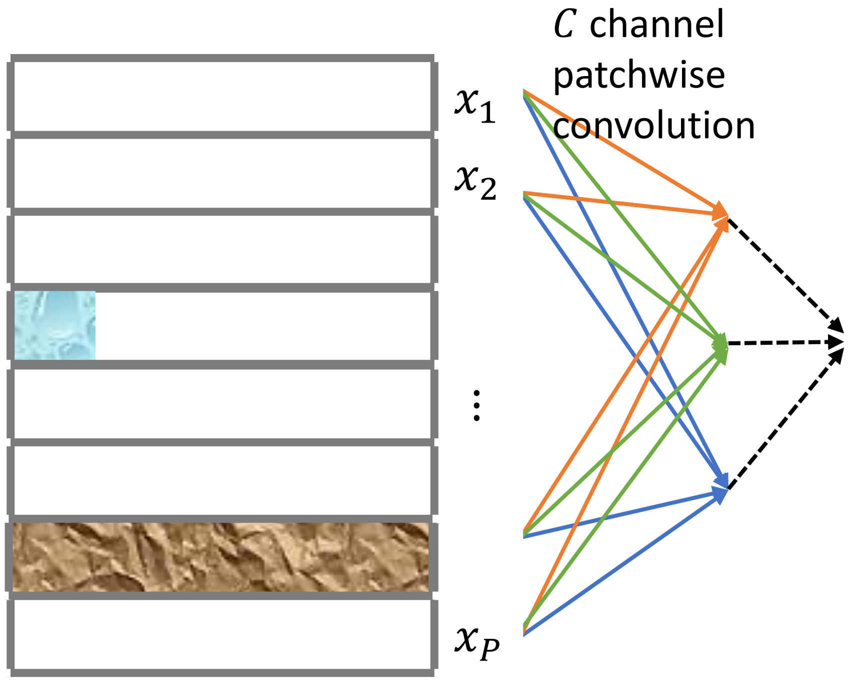
(noise/spurious feature)



One patch contains the "good" feature:
yv_k, $\mathrm{k} \in \{1, \ldots, K\}$
(ρ_k)

One patch contains the dominant "bad" feature:
$$\xi \sim \mathcal{N}\left(0, \frac{\sigma_\xi^2}{d} I\right)$$

Patchwise convolutional model

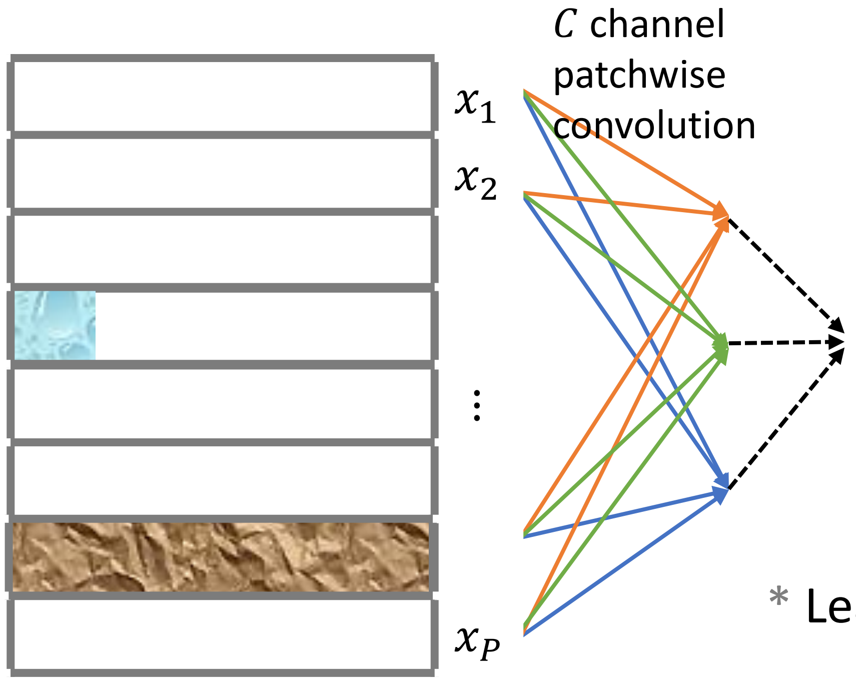


$$f(\mathbf{w}, \mathbf{x}) = \sum_c \sum_p \psi(\mathbf{x}_p \cdot \mathbf{w}_c)$$

gradient descent on logistic loss

$$L(\mathbf{w}) = \sum_{(\mathbf{x},y)\in\mathcal{D}_{\text{train}} \text{ or } \mathcal{D}_{\text{train}}^{(\text{aug})}} \log(1 + \exp(-y f(\mathbf{w}, \mathbf{x})))$$

Patchwise convolutional model



$$f(\mathbf{w},\mathbf{x}) = \sum_c \sum_p \psi(\mathbf{x}_p \cdot \mathbf{w}_c)$$

gradient descent on logistic loss

$$L(\mathbf{w}) = \sum_{(\mathbf{x},y)\in\mathcal{D}_{\text{train}} \text{ or } \mathcal{D}_{\text{train}}^{(\text{aug})}} \log(1+\exp(-yf(\mathbf{w},\mathbf{x})))$$

Fraction of datapoints with v_k

* Learning dynamic of "good" feature v_k: $\frac{d}{dt} w_c \cdot v_k \approx \rho_k \psi'(|w_c \cdot v_k|)$

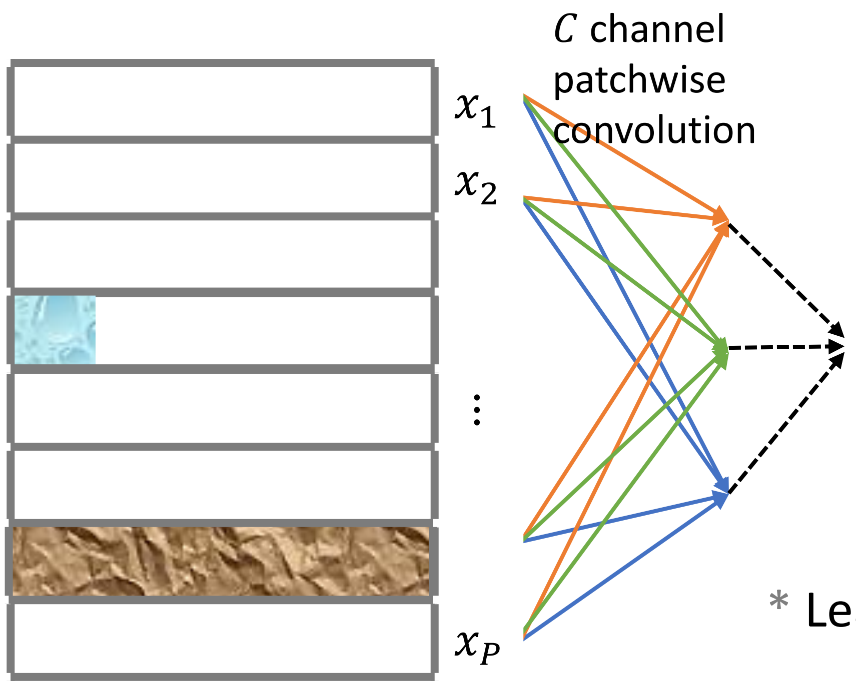
* Learning dynamic of noise $\xi^{(i)}$: $\frac{d}{dt} w_c \cdot \xi^{(i)} \approx \frac{1}{n}\sigma_\xi^2 y^{(i)} \psi'(|w_c \cdot \xi^{(i)}|)$

Noise variance

Number of datapoints

*under assumptions on feature and noise

Patchwise convolutional model



$$f(\mathbf{w}, \mathbf{x}) = \sum_c \sum_p \psi(\mathbf{x}_p \cdot \mathbf{w}_c)$$

gradient descent on logistic loss

$$L(\mathbf{w}) = \sum_{(\mathbf{x},y)\in\mathcal{D}_{\text{train}} \text{ or } \mathcal{D}_{\text{train}}^{(\text{aug})}} \log(1+\exp(-yf(\mathbf{w},\mathbf{x})))$$

Fraction of datapoints with v_k

* Learning dynamic of "good" feature v_k: $\frac{d}{dt} w_c \cdot v_k \approx \rho_k \psi'(|w_c \cdot v_k|)$

* Learning dynamic of noise $\xi^{(i)}$: $\frac{d}{dt} w_c \cdot \xi^{(i)} \approx \frac{1}{n} \sigma_\xi^2 y^{(i)} \psi'(|w_c \cdot \xi^{(i)}|)$

Noise variance

Number of datapoints

Data augmentation:

- "good" and "hard" -> "good" and "easy": Increase ρ_k of rare views k.
- "bad" and "easy" -> "bad" and "hard": Increase n (through perturbing ξ).

*under assumptions on feature and noise

Thank you