

DSC190: Machine Learning with Few Labels

Contrastive Learning
Data Manipulation

Zhiting Hu

Lecture 7, October 14, 2021

UC San Diego

HALICIOĞLU DATA SCIENCE INSTITUTE

Logistics

- In-class presentation:
 - Sign-up sheet to be released on Friday around 10am
 - At most two presentations each class

Outline

- Contrastive learning
- Data manipulation

Contrastive learning

- Take a data example x , sample a “positive” sample x_{pos} and “negative” samples x_{neg} in some way
- Then try fit a scoring model such that

$$score(x, x_{pos}) > score(x, x_{neg})$$

Contrastive learning losses: Ex 1

Learning a similarity metric discriminatively

Sample a pair of images and compute their distance:

$$D_i = \|x, x_i\|_2$$

If **positive** sample:

$$L_i = D_i^2$$



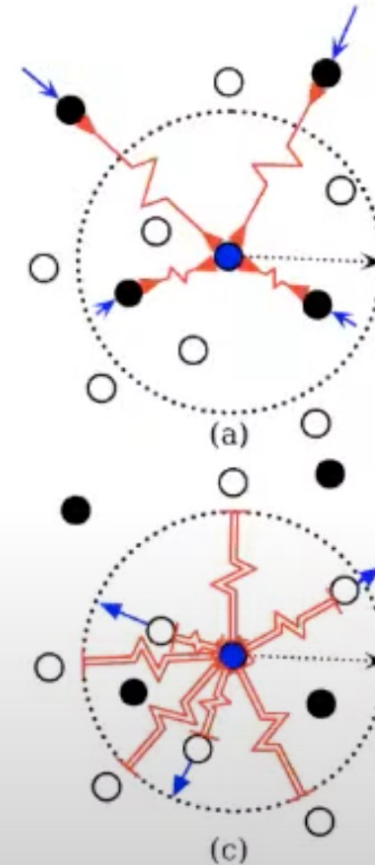
x pos

If **negative** sample:

$$L_i = \max(0, \epsilon - D_i)^2$$



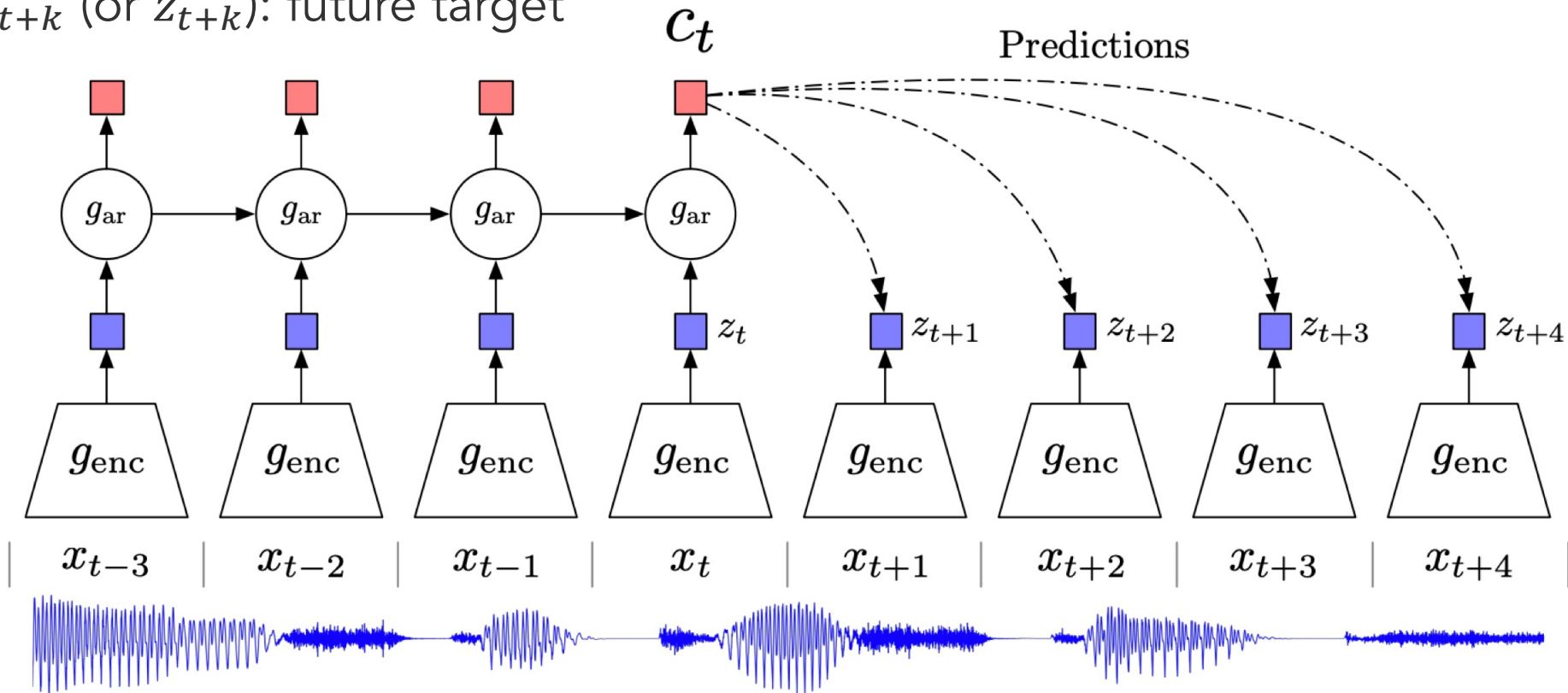
x neg



[Chopra et al., 2005; Hadsell et al., 2006]

Contrastive learning losses: Ex 2 - InfoNCE

- The CPC model
 - c_t : context representation from history
 - x_{t+k} (or z_{t+k}): future target



InfoNCE loss

- Define scoring function $f_k > 0$
- The InfoNCE loss:
 - Given $X = \{ \text{one positive sample from } p(x_{t+k} | c_t), N - 1 \text{ negative samples from the negative sampling distribution } p(x_{t+k}) \}$

$$\mathcal{L}_N = -\mathbb{E}_X \left[\log \frac{f_k(x_{t+k}, c_t)}{\sum_{x_j \in X} f_k(x_j, c_t)} \right]$$

- InfoNCE is interesting because it's effectively maximizing the **mutual information** between c_t and x_{t+k}

Mutual Information (MI)

- How much is our uncertainty about x reduced by knowing c ?

$$I(x; c) = \sum_{x,c} p(x, c) \log \frac{p(x, c)}{p(x)p(c)} = \sum_{x,c} p(x, c) \log \frac{p(x|c)}{p(x)}$$

$$= H(x) + H(c) - H(x, c)$$

$$= H(x) - H(x|c)$$

$$= KL(p(x, c) || p(x)p(c))$$

Minimizing InfoNCE \Leftrightarrow Maximizing MI

- InfoNCE loss

$$\mathcal{L}_N = -\mathbb{E}_X \left[\log \frac{f_k(x_{t+k}, c_t)}{\sum_{x_j \in X} f_k(x_j, c_t)} \right]$$

- The loss is optimized when

$$f_k(x_{t+k}, c_t) \propto \frac{p(x_{t+k} | c_t)}{p(x_{t+k})}$$

- Proof:

$$\begin{aligned} p(\text{sample } i \text{ is positive} | X, c_t) &= \frac{p(x_i | c_t) \prod_{l \neq i} p(x_l)}{\sum_{j=1}^N p(x_j | c_t) \prod_{l \neq j} p(x_l)} \\ &= \frac{\frac{p(x_i | c_t)}{p(x_i)}}{\sum_{j=1}^N \frac{p(x_j | c_t)}{p(x_j)}} \end{aligned}$$

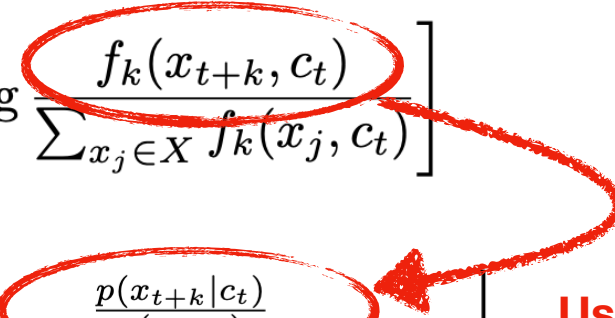
- How does this loss maximize the mutual information?

$$\mathcal{L}_N = -\mathbb{E}_X \left[\log \frac{f_k(x_{t+k}, c_t)}{\sum_{x_j \in X} f_k(x_j, c_t)} \right]$$

- How does this loss maximize the mutual information?

$$\mathcal{L}_N = -\mathbb{E}_X \left[\log \frac{f_k(x_{t+k}, c_t)}{\sum_{x_j \in X} f_k(x_j, c_t)} \right]$$
$$\mathcal{L}_N^{\text{opt}} = -\mathbb{E}_X \log \left[\frac{\frac{p(x_{t+k}|c_t)}{p(x_{t+k})}}{\frac{p(x_{t+k}|c_t)}{p(x_{t+k})} + \sum_{x_j \in X_{\text{neg}}} \frac{p(x_j|c_t)}{p(x_j)}} \right]$$

Use proportionality condition



- How does this loss maximize the mutual information?

$$\mathcal{L}_N = -\mathbb{E}_X \left[\log \frac{f_k(x_{t+k}, c_t)}{\sum_{x_j \in X} f_k(x_j, c_t)} \right]$$

$$\begin{aligned} \mathcal{L}_N^{\text{opt}} &= -\mathbb{E}_X \log \left[\frac{\frac{p(x_{t+k}|c_t)}{p(x_{t+k})}}{\frac{p(x_{t+k}|c_t)}{p(x_{t+k})} + \sum_{x_j \in X_{\text{neg}}} \frac{p(x_j|c_t)}{p(x_j)}} \right] \\ &= \mathbb{E}_X \log \left[1 + \frac{p(x_{t+k})}{p(x_{t+k}|c_t)} \sum_{x_j \in X_{\text{neg}}} \frac{p(x_j|c_t)}{p(x_j)} \right] \end{aligned}$$

Take -ve inside log

- How does this loss maximize the mutual information?

$$\mathcal{L}_N = -\mathbb{E}_X \left[\log \frac{f_k(x_{t+k}, c_t)}{\sum_{x_j \in X} f_k(x_j, c_t)} \right]$$

$$\begin{aligned} \mathcal{L}_N^{\text{opt}} &= -\mathbb{E}_X \log \left[\frac{\frac{p(x_{t+k}|c_t)}{p(x_{t+k})}}{\frac{p(x_{t+k}|c_t)}{p(x_{t+k})} + \sum_{x_j \in X_{\text{neg}}} \frac{p(x_j|c_t)}{p(x_j)}} \right] \\ &= \mathbb{E}_X \log \left[1 + \frac{p(x_{t+k})}{p(x_{t+k}|c_t)} \sum_{x_j \in X_{\text{neg}}} \frac{p(x_j|c_t)}{p(x_j)} \right] \\ &\approx \mathbb{E}_X \log \left[1 + \frac{p(x_{t+k})}{p(x_{t+k}|c_t)} (N-1) \mathbb{E}_{x_j} \frac{p(x_j|c_t)}{p(x_j)} \right] \end{aligned}$$

This approximation becomes more accurate as N increases, so it is preferable to use large negative samples

- How does this loss maximize the mutual information?

$$\mathcal{L}_N = -\mathbb{E}_X \left[\log \frac{f_k(x_{t+k}, c_t)}{\sum_{x_j \in X} f_k(x_j, c_t)} \right]$$

$$\mathcal{L}_N^{\text{opt}} = -\mathbb{E}_X \log \left[\frac{\frac{p(x_{t+k}|c_t)}{p(x_{t+k})}}{\frac{p(x_{t+k}|c_t)}{p(x_{t+k})} + \sum_{x_j \in X_{\text{neg}}} \frac{p(x_j|c_t)}{p(x_j)}} \right]$$

$$= \mathbb{E}_X \log \left[1 + \frac{p(x_{t+k})}{p(x_{t+k}|c_t)} \sum_{x_j \in X_{\text{neg}}} \frac{p(x_j|c_t)}{p(x_j)} \right]$$

$$\approx \mathbb{E}_X \log \left[1 + \frac{p(x_{t+k})}{p(x_{t+k}|c_t)} (N-1) \mathbb{E}_{x_j} \frac{p(x_j|c_t)}{p(x_j)} \right] = 1$$

$$= \mathbb{E}_X \log \left[1 + \frac{p(x_{t+k})}{p(x_{t+k}|c_t)} (N-1) \right]$$

- How does this loss maximize the mutual information?

$$\mathcal{L}_N = -\mathbb{E}_X \left[\log \frac{f_k(\mathbf{x}_{t+k}, \mathbf{c}_t)}{\sum_{\mathbf{x}_j \in X} f_k(\mathbf{x}_j, \mathbf{c}_t)} \right]$$

$$\begin{aligned} \mathcal{L}_N^{\text{opt}} &= -\mathbb{E}_X \log \left[\frac{\frac{p(\mathbf{x}_{t+k} | \mathbf{c}_t)}{p(\mathbf{x}_{t+k})}}{\frac{p(\mathbf{x}_{t+k} | \mathbf{c}_t)}{p(\mathbf{x}_{t+k})} + \sum_{\mathbf{x}_j \in X_{\text{neg}}} \frac{p(\mathbf{x}_j | \mathbf{c}_t)}{p(\mathbf{x}_j)}}} \right] \\ &= \mathbb{E}_X \log \left[1 + \frac{p(\mathbf{x}_{t+k})}{p(\mathbf{x}_{t+k} | \mathbf{c}_t)} \sum_{\mathbf{x}_j \in X_{\text{neg}}} \frac{p(\mathbf{x}_j | \mathbf{c}_t)}{p(\mathbf{x}_j)} \right] \\ &\approx \mathbb{E}_X \log \left[1 + \frac{p(\mathbf{x}_{t+k})}{p(\mathbf{x}_{t+k} | \mathbf{c}_t)} (N-1) \mathbb{E}_{\mathbf{x}_j} \frac{p(\mathbf{x}_j | \mathbf{c}_t)}{p(\mathbf{x}_j)} \right] \\ &= \mathbb{E}_X \log \left[1 + \frac{p(\mathbf{x}_{t+k})}{p(\mathbf{x}_{t+k} | \mathbf{c}_t)} (N-1) \right] \\ &\geq \mathbb{E}_X \log \left[\frac{p(\mathbf{x}_{t+k})}{p(\mathbf{x}_{t+k} | \mathbf{c}_t)} N \right] \\ &= -I(\mathbf{x}_{t+k}, \mathbf{c}_t) + \log(N), \end{aligned}$$

- How does this loss maximize the mutual information?

$$\mathcal{L}_N = -\mathbb{E}_X \left[\log \frac{f_k(x_{t+k}, c_t)}{\sum_{x_j \in X} f_k(x_j, c_t)} \right]$$

$$I(x_{t+k}, c_t) \geq \log(N) - \mathcal{L}_N$$

Summary so far: Contrastive learning

- Contrastive learning is a way of doing self-supervised learning
- Mutual information

$$\begin{aligned} I(x; c) &= \sum_{x,c} p(x, c) \log \frac{p(x, c)}{p(x)p(c)} = \sum_{x,c} p(x, c) \log \frac{p(x|c)}{p(x)} \\ &= H(x) + H(c) - H(x, c) \\ &= H(x) + H(x|c) \\ &= KL(p(x, c) || p(x)p(c)) \end{aligned}$$

- InfoNCE \Leftrightarrow MI

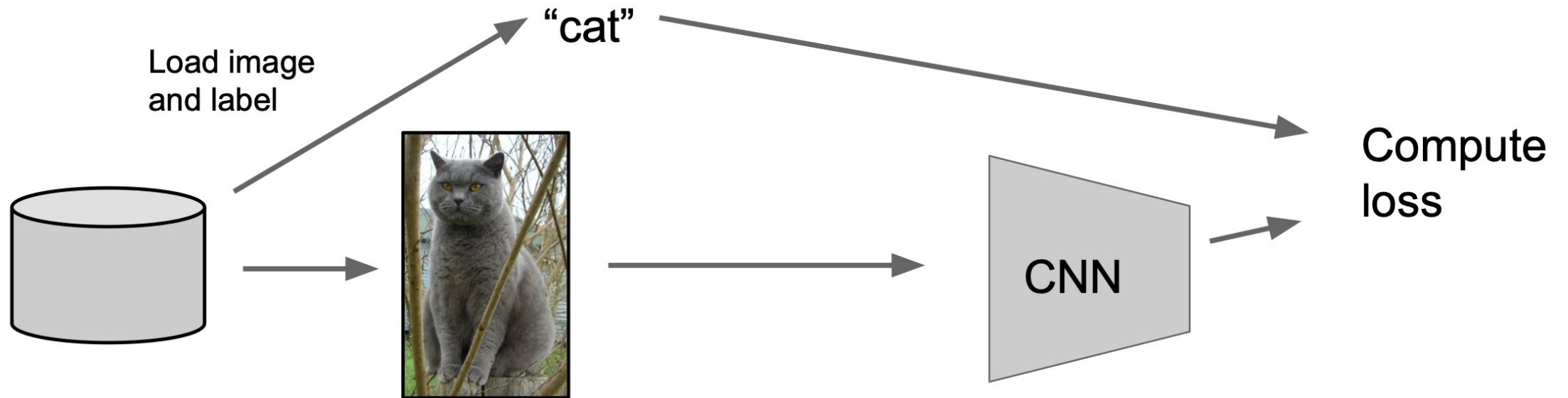
Data Manipulation

Data manipulation

- Data augmentation
 - Applies label-preserving transformations on original data points to expand the data size
- Data reweighting
 - Assigns an importance weight to each instance to adapt its effect on learning
- Data synthesis
 - Generates entire artificial examples
- Curriculum learning
 - Makes use of data instances in an order based on “difficulty”
- ...

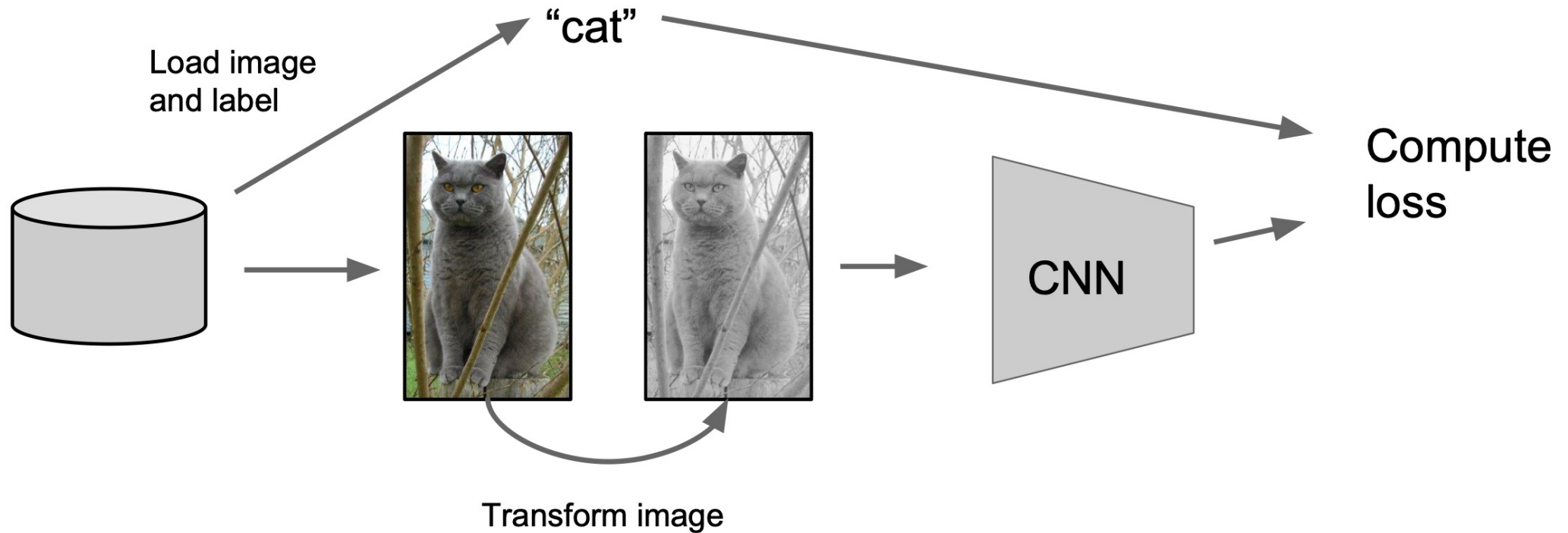
Data augmentation

- Applies **label-preserving transformations** on original data points to expand the data size



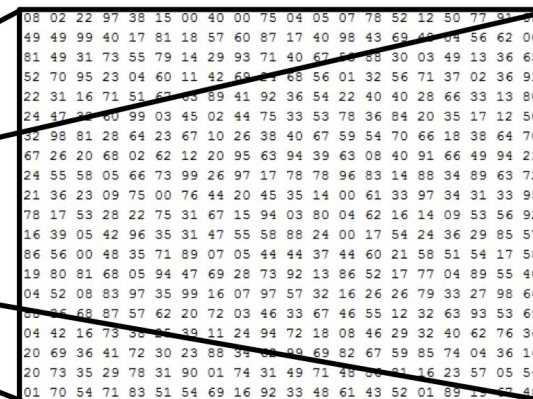
Data augmentation

- Applies **label-preserving transformations** on original data points to expand the data size



Data augmentation for image

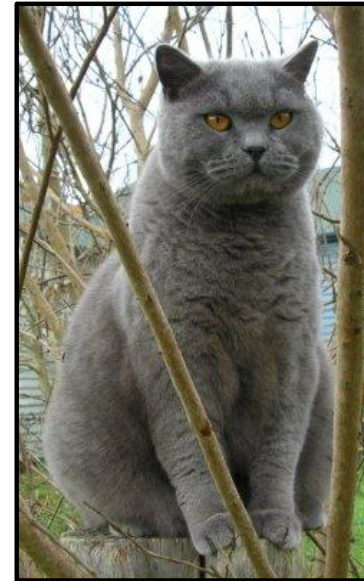
- Change the pixels without changing the label
- Train on transformed data
- VERY widely used



What the computer sees

Data augmentation for image

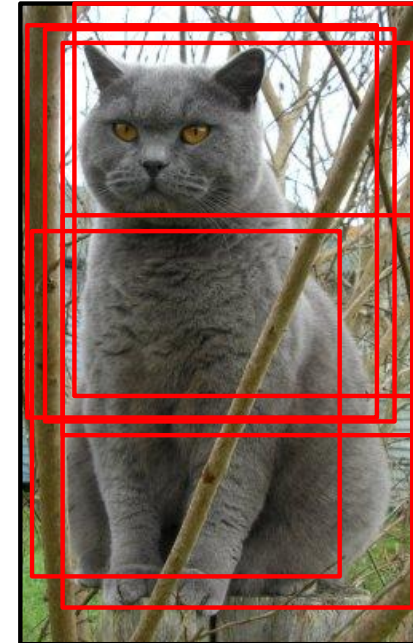
1. Horizontal flips



Data augmentation for image

2. Random crops/scales

Training: sample random crops / scales



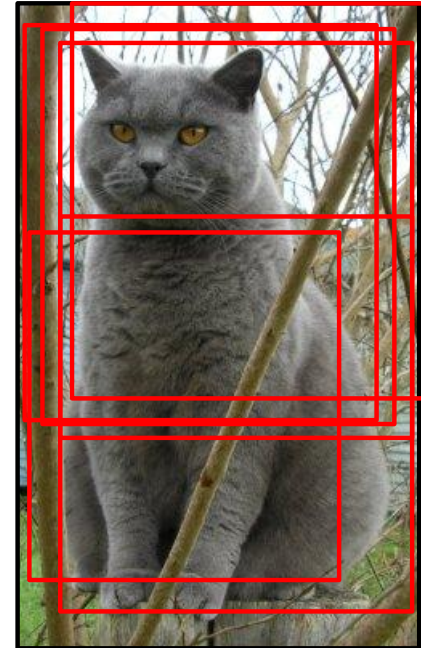
Data augmentation for image

2. Random crops/scales

Training: sample random crops / scales

ResNet:

1. Pick random L in range $[256, 480]$
2. Resize training image, short side = L
3. Sample random 224×224 patch



Data augmentation for image

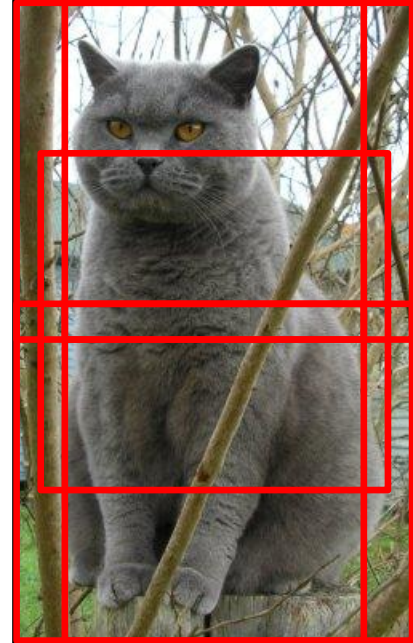
2. Random crops/scales

Training: sample random crops / scales

ResNet:

1. Pick random L in range $[256, 480]$
2. Resize training image, short side = L
3. Sample random 224×224 patch

Testing: average a fixed set of crops



Data augmentation for image

2. Random crops/scales

Training: sample random crops / scales

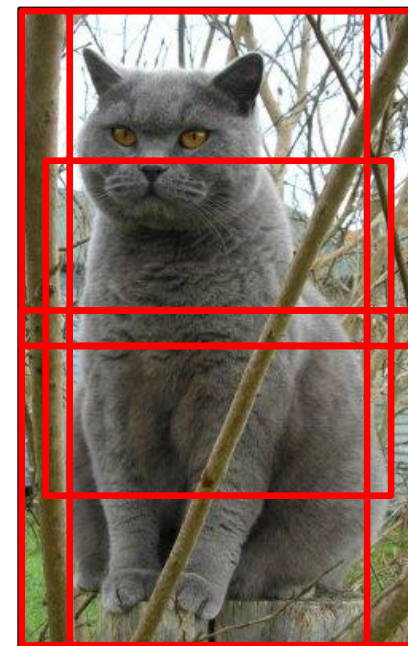
ResNet:

1. Pick random L in range $[256, 480]$
2. Resize training image, short side = L
3. Sample random 224×224 patch

Testing: average a fixed set of crops

ResNet:

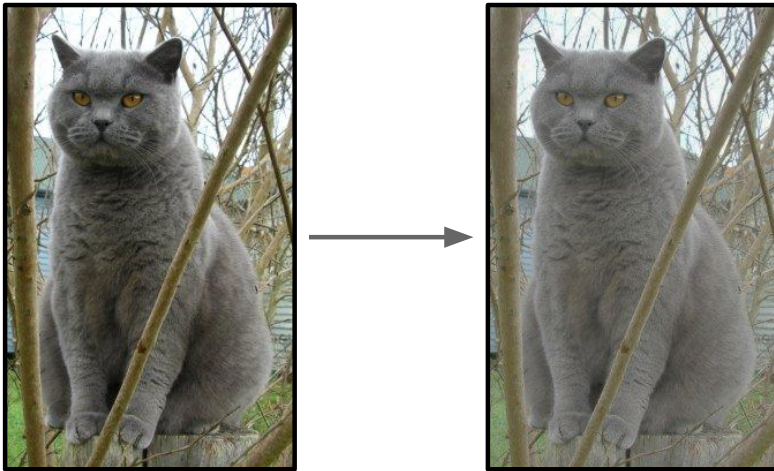
1. Resize image at 5 scales: $\{224, 256, 384, 480, 640\}$
2. For each size, use 10 224×224 crops: 4 corners + center, + flips



Data augmentation for image

3. Color jitter

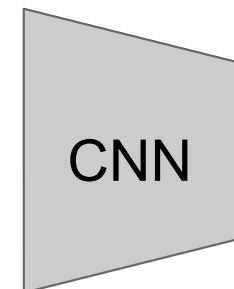
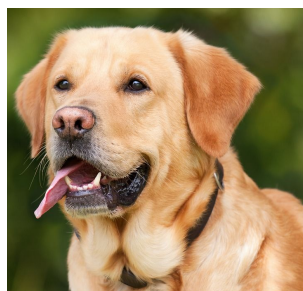
Randomly jitter contrast



Data augmentation for image

4. Mixup

- **Training:** Train on random blends of images
- **Testing:** Use original images



Target label:
cat: 0.4
dog: 0.6

Randomly blend the pixels of pairs of training images, e.g. 40% cat, 60% dog

[Zhang et al., “*mixup*: Beyond Empirical Risk Minimization”, ICLR 2018]

Data augmentation for image

5. Get creative!

Random mix/combinations of :

- translation
- rotation
- stretching
- shearing
- lens distortions, ...

Data augmentation for text

Methods	Level	Diversity	Tasks	Related Work
Synonym replacement	Token	Low	Text classification Sequence labeling	Kolomiyets et al. (2011), Zhang et al. (2015a), Yang (2015), Miao et al. (2020), Wei and Zou (2019)
Word replacement via LM	Token	Medium	Text classification Sequence labeling Machine translation	Kolomiyets et al. (2011), Gao et al. (2019) Kobayashi (2018), Wu et al. (2019a) Fadaee et al. (2017)
Random insertion, deletion, swapping	Token	Low	Text classification Sequence labeling Machine translation Dialogue generation	Iyyer et al. (2015), Xie et al. (2017) Artetxe et al. (2018), Lample et al. (2018) Xie et al. (2020), Wei and Zou (2019)
Compositional Augmentation	Token	High	Semantic Parsing Sequence labeling Language modeling Text generation	Jia and Liang (2016) , Andreas (2020) Nye et al. (2020), Feng et al. (2020) Furrer et al. (2020) , Guo et al. (2020)
Paraphrasing	Sentence	High	Text classification Machine translation Question answering Dialogue generation Text summarization	Yu et al. (2018), Xie et al. (2020) Chen et al. (2019), He et al. (2020) Chen et al. (2020c), Cai et al. (2020)
Conditional generation	Sentence	High	Text classification Question answering	Anaby-Tavor et al. (2020), Kumar et al. (2020) Zhang and Bansal (2019), Yang et al. (2020)

Data augmentation for text

White-box attack	Token or Sentence	Medium	Text classification Sequence labeling Machine translation	Miyato et al. (2017), Ebrahimi et al. (2018b) Ebrahimi et al. (2018a), Cheng et al. (2019), Chen et al. (2020d)
Black-box attack	Token or Sentence	Medium	Text classification Sequence labeling Machine translation Textual entailment Dialogue generation Text Summarization	Jia and Liang (2017) Belinkov and Bisk (2017), Zhao et al. (2017) Ribeiro et al. (2018), McCoy et al. (2019) Min et al. (2020), Tan et al. (2020)
Hidden-space perturbation	Token or Sentence	High	Text classification Sequence labeling Speech recognition	Hsu et al. (2017), Hsu et al. (2018) Wu et al. (2019b), Chen et al. (2021) Malandrakis et al. (2019), Shen et al. (2020)
Interpolation	Token	High	Text classification Sequence labeling Machine translation	Miao et al. (2020), Chen et al. (2020c) Cheng et al. (2020b), Chen et al. (2020a) Guo et al. (2020)

Data reweighting

- Assigns an importance weight to each instance to adapt its effect on learning
 - Weighting by inverse class frequency
 - Weighting by the magnitude of loss

$$\min_{\theta} - \mathbb{E}_{x_i \sim \mathcal{D}} [\phi_i \log p_{\theta}(x_i)]$$

Automatically learn the data weights

- Can we learn ϕ_i automatically?

$$\min_{\theta} - \mathbb{E}_{x_i \sim \mathcal{D}} [\phi_i \log p_{\theta}(x_i)]$$

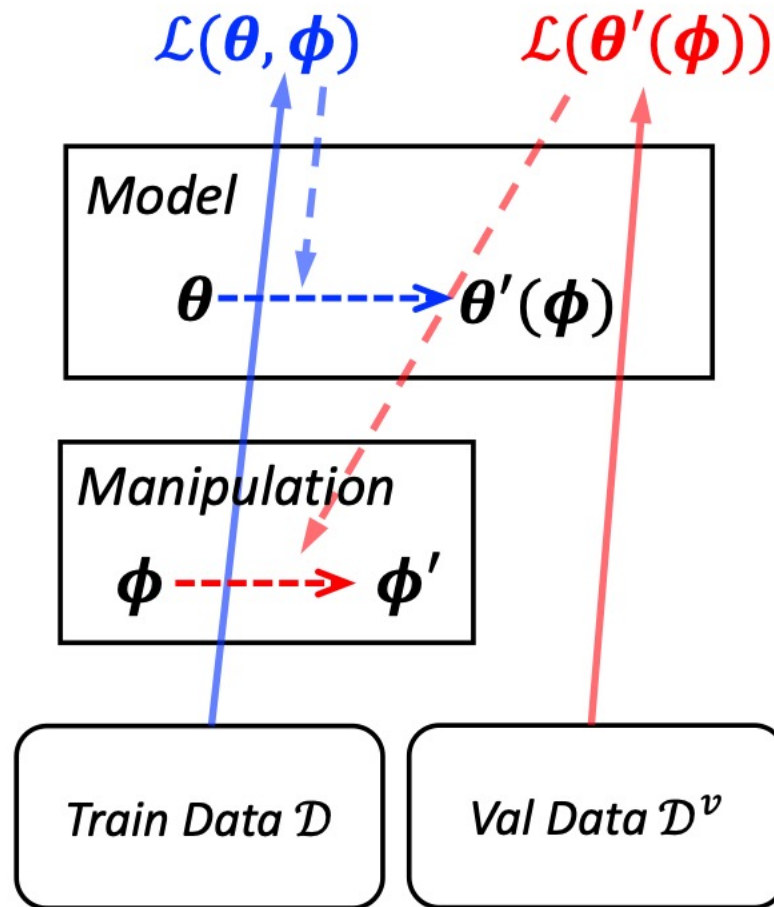
- Training set \mathcal{D} , a held-out “validation” set \mathcal{D}_v
- Intuition: after training the model θ on the weighted data, the model gets better performance on the validation set

$$\theta' = \operatorname{argmin}_{\theta} - \mathbb{E}_{x_i \sim \mathcal{D}} [\phi_i \log p_{\theta}(x_i)]$$

- θ' is a function of ϕ , i.e., $\theta' = \theta'(\phi)$

$$\phi' = \operatorname{argmin}_{\phi} - \mathbb{E}_{x_i \sim \mathcal{D}_v} [\log p_{\theta'(\phi)}(x_i)]$$

Automatically learn the data weights



Apply the same algorithm to learn data augmentation

- Augmentation function $x' = g_\phi(x)$. Can we learn ϕ automatically?

$$\min_{\theta} - \mathbb{E}_{x_i \sim \mathcal{D}} \left[\log p_{\theta}(g_{\phi}(x_i)) \right]$$

- Training set \mathcal{D} , a held-out “validation” set \mathcal{D}_v
- Intuition: after training the model θ on the augmented data, the model gets better performance on the validation set

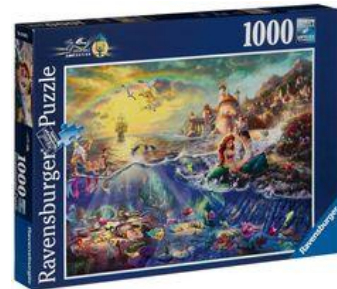
$$\theta' = \operatorname{argmin}_{\theta} - \mathbb{E}_{x_i \sim \mathcal{D}} \left[\log p_{\theta}(g_{\phi}(x_i)) \right]$$

- θ' is a function of ϕ , i.e., $\theta' = \theta'(\phi)$

$$\phi' = \operatorname{argmin}_{\phi} - \mathbb{E}_{x_i \sim \mathcal{D}_v} \left[\log p_{\theta'(\phi)}(x_i) \right]$$

Curriculum learning

NOT MY FIRST JIGSAW PUZZLE



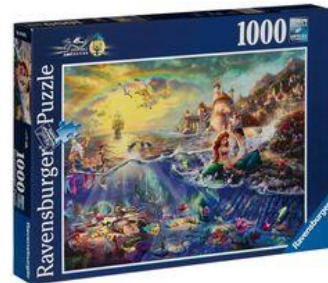
Curriculum learning

MY FIRST JIGSAW PUZZLE



Curriculum learning

LEARNING COGNITIVE TASKS (CURRICULUM):



Curriculum learning

- Standard supervised learning:
 - Data is sampled randomly
- Curriculum learning:
 - Instead of randomly selecting training points, select easier examples first, slowly exposing the more difficult examples from easiest to the most difficult
 - Key: definition of “difficulty”

Curriculum learning

- (Bengio et al, 2009): setup of paradigm, object recognition of geometric shapes using a perceptron; *difficulty is determined by user from geometric shape*



- (Zaremba 2014): LSTMs used to evaluate short computer programs; *difficulty is automatically evaluated from data – nesting level of program.*
- (Amodei et al, 2016): End-to-end speech recognition in english and mandarin; *difficulty is automatically evaluated from utterance length.*
- (Jesson et al, 2017): deep learning segmentation and detection; *human teacher (user/programmer) determines difficulty.*

Key Takeaways

- Contrastive learning is a way of doing self-supervised learning

- Mutual information

$$I(x; c) = \sum_{x,c} p(x, c) \log \frac{p(x, c)}{p(x)p(c)}$$

- Data manipulation
 - Augmentation
 - Reweighting
 - Synthesis (later)
 - Curriculum learning

Questions?