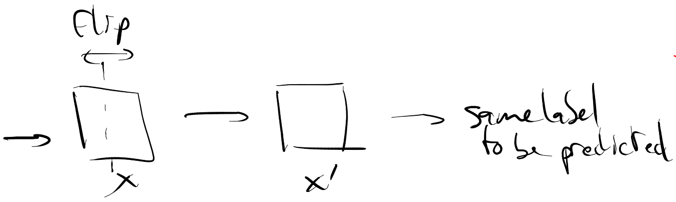
Forms of weak supervision

I Small data

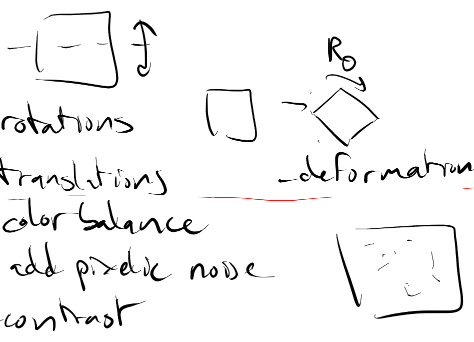
Data augmentation

ex: image classification task
add transformations

label: invariant, equivariant



- rotations
- translations
- deformations
- color balance
- add pixelic noise
- contrast



geometric transforms

value-based transforms

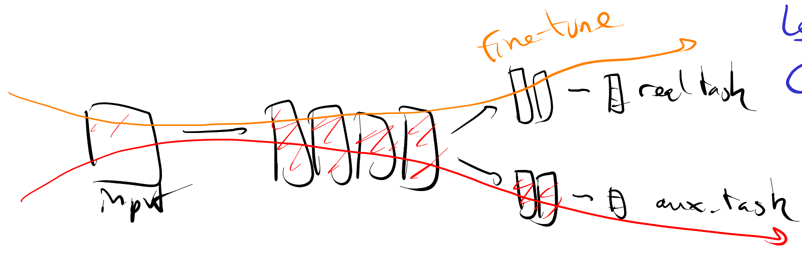
- use a simulator
  - generate lots of data
  - how realistic?

Multi-tasking

- one real task + one auxiliary task

real task: few labeled data

auxiliary task: lots of labeled data



Loss:

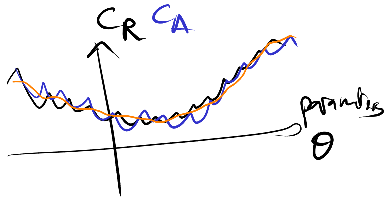
C_R

C_A

tasks need to be close

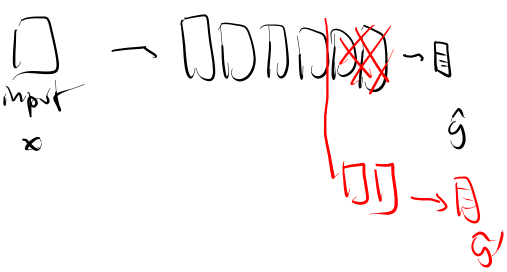
supervised with lots of data

Loss: $\dfrac{C_R + C_A}{2}$



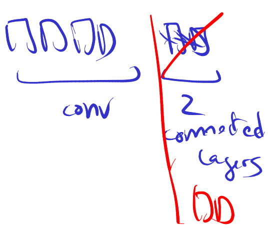
Transfer learning

- sequential training: first on auxiliary task ← pick a pre-trained model
  then on real task



pre-trained model

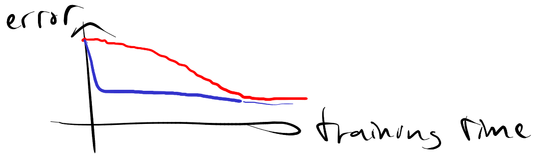
conv: ResNet / VGG
ImageNet

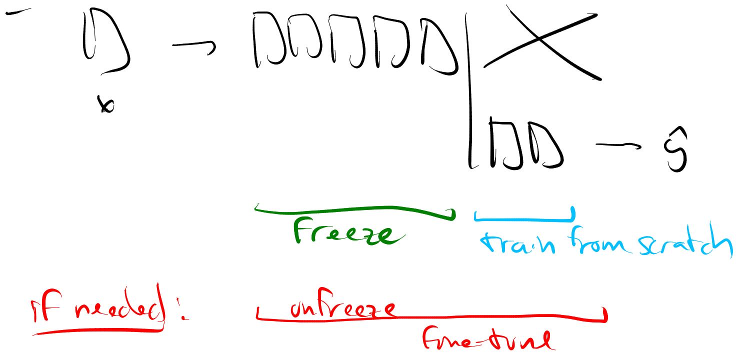


- analysis from [Rethinking ImageNet pre-training]

small data:
helps in getting good features

big data:
≠ from training from scratch,
a big boost in training time

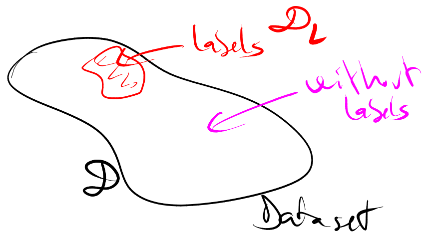




II Forms of weak supervision

Amount of data, but few are labeled

Semi-supervision:

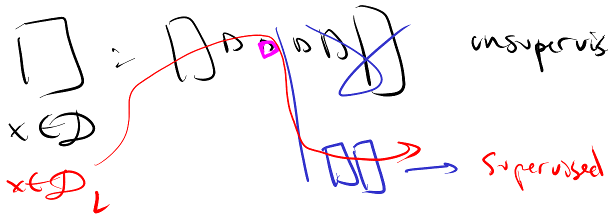


ex: when labeling is costly
(requires time, expertise, ...)

Several approaches:

1) unsupervised training on full dataset $\mathcal{D}$ → good representation → supervised task on $\mathcal{D}_L$

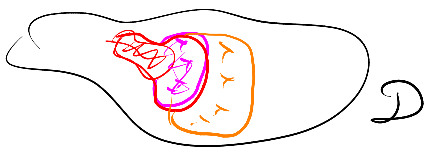
ex: auto encoder



2) supervised training $\mathcal{D}_L$ → label some of the unlabeled samples → bigger dataset

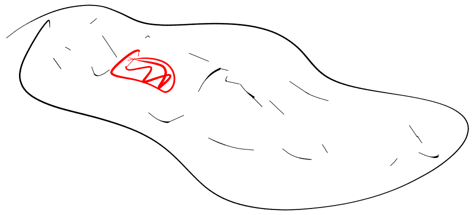
$\mathcal{D}_{new} = \mathcal{D}_L \cup \mathcal{D}_{\substack{new\\labels}}$

iterate



Issue: what if mistakes?

3) supervised training $\mathcal{D}_L$ → apply to full dataset $\mathcal{D}$ → check some properties & adjust parameters



eg: global bias

target: A / B

50% 50%

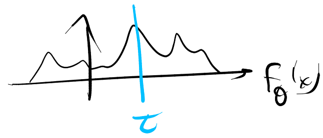
30% 70%

↓

adjust the decision threshold

$f_\theta(x) <> 0$

$f_\theta(x) <> \tau$



properties: - bias
- density
- margin (SVM)

Weak supervision

↳ more general:

eg: labels could be noisy

self-supervision

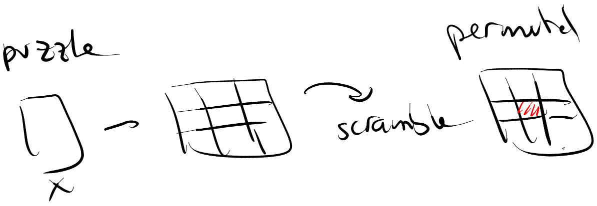
↳ used for pretraining

→ no supervision (no label given by hand)

↳ unsupervised task formulated as a supervised one with automatic labels

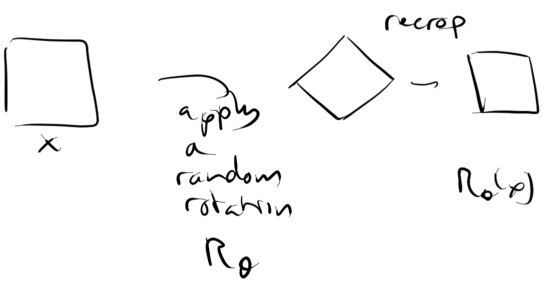
eg: image classification

- image puzzle



→ dummy task: where was this middle piece? classification on 9 classes

- image rotations



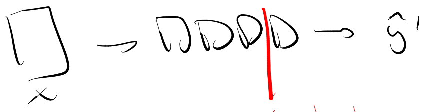
task → what was the angle θ? regression task

eg: video classification

- predict next frame
- give 3 frames: ask temporal order

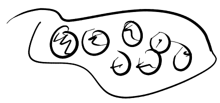
Building on teacher-student techniques

- "ClusterFit":



dummy task

↳ clustering → add labels



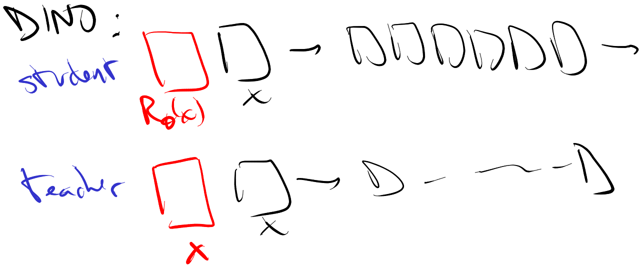
teacher

train a new network:

task: predict this label, i.e. which cluster x belongs to

student

- DINO:



randomly initialized

average of the past students (recent history) moving average

data augmentation → learning to be invariant to the chosen group of transforms

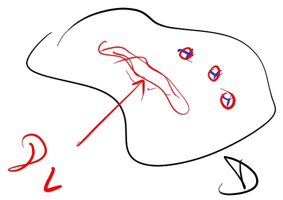
+ something against collapse

↳ just train a linear classifier on top ⇒ almost as good as best supervised technique ever on ImageNet

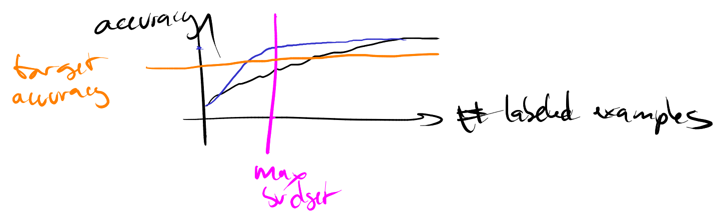
Active learning

same setting as semi-supervision + ask some samples to be labeled

↳ costly labeling

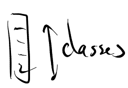


goal: increase global accuracy as fast as possible (in terms of # of labeled samples)



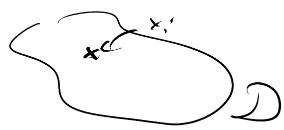
- large dataset: $D = \{x_i\}$
- labels for now: D_L : $(y_1, \ldots, y_p)$ with $p \ll |D|$
- which $x_i \in D$ to pick? ($i \in]p, |D|]$) to ask to be labeled

* apply the current model f_θ to all samples → predictions $\hat{y}_i = (y_i^c)_{c \in C}$ if classification task



Local methods

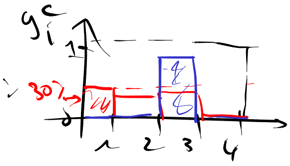
→ quantify the impact of the choice of x_i on the prediction for that point only



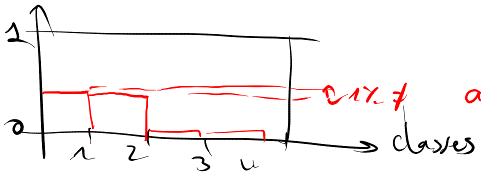
uncertainty: pick x_i for which f_θ is the most uncertain

$$\underset{x \in \mathcal{D} \setminus \mathcal{D}_L}{\arg\min} \ \sup_{c \in C} \hat{y}_i^c$$

For x_i:



margin:



$$\underset{x_i \in \mathcal{D} \setminus \mathcal{D}_L}{\arg\min} \ \hat{y}_i^{c_1} - \hat{y}_i^{c_2}$$

where
$c_1 = \arg\max_c \hat{y}_i^c$
c_2 = second argmax

entropy: $H(\hat{y}_i) = -\sum_{c \in C} \hat{y}_i^c \log \hat{y}_i^c$

$$\underset{i}{\arg\max} \ H(\hat{y}_i)$$

↳ quantifies how spread the distribution $\hat{y}_i$ over classes is

pick the most uniform

query by committee:

if predictor = ensemble of K models ↳ $\hat{y}_{i,k}$

↳ do models agree? pick i where models disagree most

Global methods

↳ quantify impact of the sample choice over all dataset examples

Expected model change

$f_\theta \xrightarrow[\text{new labeled sample}]{\text{retrain with}} f_{\theta+\delta\theta} \to$ apply to all unlabeled samples $\Rightarrow$ impact?

just one training step ($\nabla\downarrow$)

$$\theta_t \longrightarrow \theta_{t+1} = \theta_t - \eta \nabla_\theta \text{Loss}(\hat{y}_i, \delta_{c^*})$$

↑ true label

quantify the information gain as $\|\theta_{t+1} - \theta_t\|$

$$\propto \|\nabla_\theta \text{Loss}(\hat{y}_i, \delta_{c^*})\|$$

true label c^* not known $\Rightarrow$ average over possibilities

$$\mathbb{E}_{c^* \in C}\left[\|\nabla_\theta \text{Loss}(\hat{y}_i, \delta_{c^*})\|\right]$$

↳ $p(c^*) = \hat{y}_i^{c^*}$

$$\sum_{c \in C} \hat{y}_i^c \|\nabla_\theta \text{Loss}(\hat{y}_i^c, \delta_c)\|$$

Expected error reduction

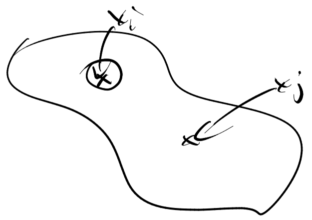
density-based method

$$\arg\min_i \sum_{c \in C} \hat{y}_i^c \sum_j \text{prediction variation for sample } x_j$$

all samples

if trained also with (x_i, c)

or similarity between x_i & x_j



$$\theta_t \to \theta_{t+1} \quad \delta\theta = -\eta \nabla_\theta \text{loss}(x_i, c)$$

$$f_{\theta_{t+1}}(x_j) = f_{\theta_t}(x_j) + \underbrace{\boxed{\delta\theta} \cdot \nabla_\theta f_{\theta_t}(x_j)}_{\text{prediction variation}} + O(\delta\theta^2)$$

$$-\eta \nabla_\theta \text{Loss}(x_i, c) \cdot \nabla_\theta f_{\theta_t}(x_j)$$

$$\nabla_\theta f_\theta(x_i) \frac{\partial L}{\partial f_\theta(x_i)} \nabla_\theta f_{\theta_t}(x_j)$$

(chain rule)

$\simeq$ influence functions

$$k(x_i, x_j) = \nabla_\theta f_\theta(x_i) \cdot \nabla_\theta f_\theta(x_j)$$

similarity

Cons: computional power

Pros: avoid choosing outliers

but focus on yet-unlabeled clusters of similar points