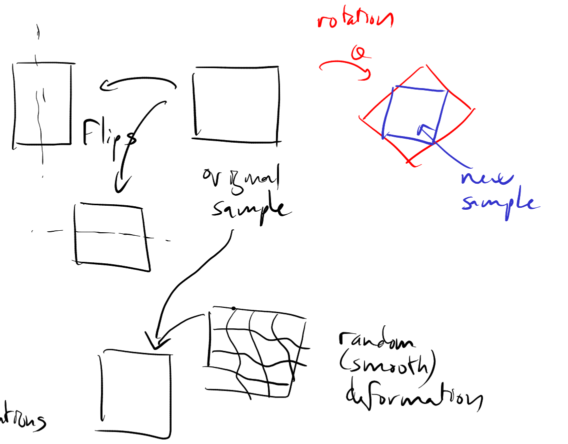
Small data : weak supervision, transfer & incorporation of priors

I Dealing with small data

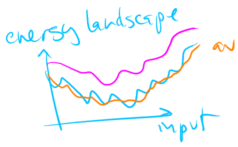
Data augmentation

- For image classification tasks :
  - deformation
  - add rotations } spatial domain
  - flips
  - contrast
  - color balance } intensity changes
  - noise ...

  $\underbrace{\hspace{6em}}$ model the "noise" non-meaningful transformations



- use a simulator
  ↳ produce quantities of input data

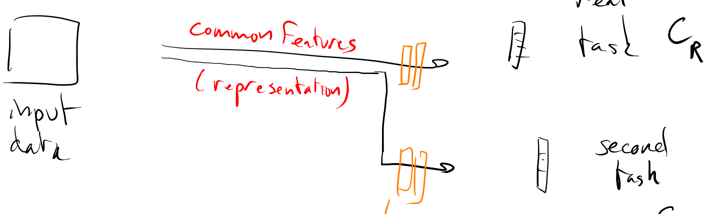


Multi-tasking

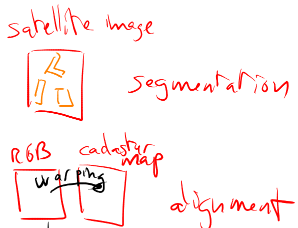
- consider (at the same training time) another task

Settings

data
- small labeled data
- lot of unlabeled data



second task : not too unrelated to the real task

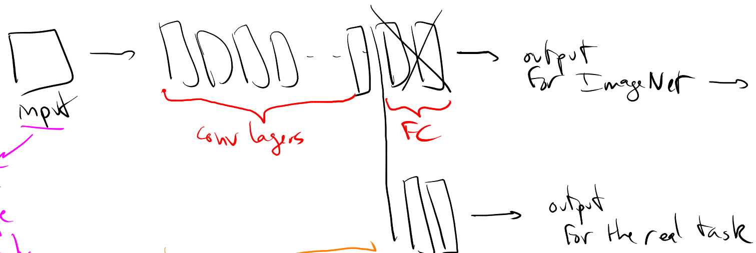


Loss : $(C_R + C_S)/2$

Transfer learning

- sequential training : - first C_S ← pick a pre-trained network
  - second C_R

  eg : for computer vision task, pre-trained { VGG on ImageNet / ResNet



input : of the same type for both tasks (or adapt)
- might need pre-processing (scaling, etc.)
- # input channels

train only this → possible second step : unfreeze all weights

(high-level) "features" ⟷ (simple) classifier

- analysis from [Rethinking ImageNet pre-training]
  - small data : help in getting good features
  - big data : ≠ with training from scratch : big boost in training time (not necessary accuracy gain)

II Forms of weak supervision

→ Few labeled examples

Semi-supervision

D_L labeled samples (small part)

D_U dataset $\mathcal{F}$

eg : when labeling is costly (requires time, expert ...)

several approaches;

1) unsupervised training (on full set) $\mathcal{F}$ → good representation → supervised training $\mathcal{D}_\mathcal{L}$

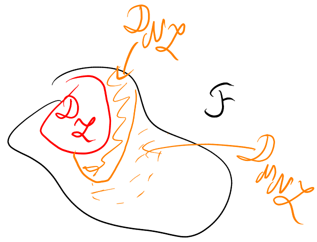
↳ features
↳ clustering

↳ typically: auto-encoders

2) supervised training $\mathcal{D}_\mathcal{L}$ → label some of the unlabeled samples $\mathcal{D}_\mathcal{U}$ → bigger training set $\mathcal{D}_\mathcal{L} \cup \mathcal{D}_{N\mathcal{L}}$

↳ $\mathcal{D}_{N\mathcal{L}}$ newly labeled

iterate



issues: if mistakes, learn from wrongly labeled data

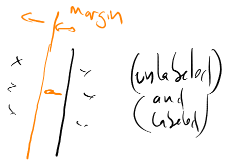
3) supervised training $\mathcal{D}_\mathcal{L}$ → apply to full dataset $\mathcal{F}$ → check some properties

ex: bias 50% / 50%

↳ on $\mathcal{F}$: 60% / 60%

↳ correct bias on $\mathcal{F}$

- density
- margin



Weak supervision

↳ more general: ex: labels could be noisy

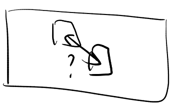
Self-supervision

→ unsupervised (pre)training

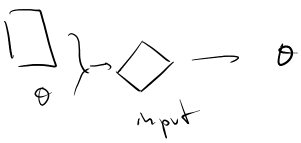
↳ supervised task (on $\mathcal{F}$) → with a fake task with labels for all samples in $\mathcal{F}$

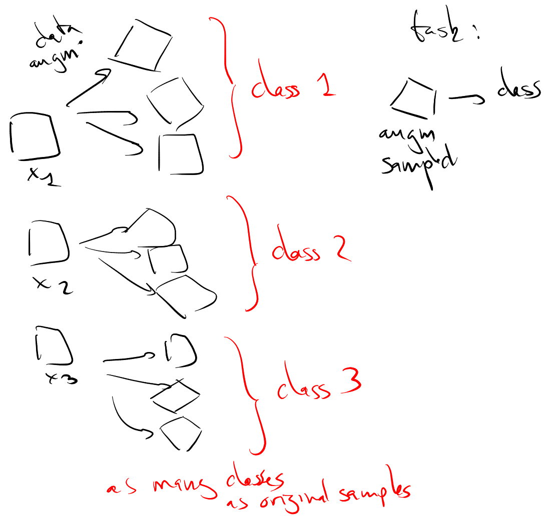
ex: image classification

→ image puzzle : extract patches from same image & ask for geometrical relation



→ add a rotation to the image → task: retrieve the angle θ (random)





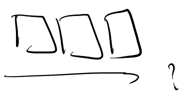
→ { data augmentation
define "classes";
1 class = {all augmented input data coming from same sample}

ex: video classification

↳ auxiliary task:

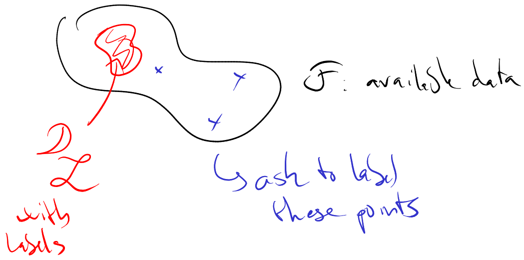
→ predict next frame : fully supervised

→ give 3 frames, ask whether temporal order is correct

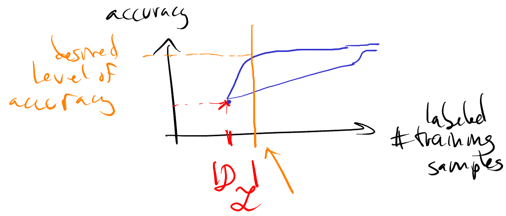


Active learning

Same setting as semi-supervision, + ask some samples to be labeled
(eg: costly labeling)



goal: increase accuracy as fast as possible
(as a function of number of samples seen)



- large dataset $\mathcal{F} = \{x_i\}$
- labels for few: $(y_1, \dots y_p)$ with $p \ll |\mathcal{F}|$ ($\mathcal{D}_\mathcal{L}$)
- which x_i (with $i \in]p,n]$) to pick? to be asked for labels

$\rightarrow$ apply current model F to all samples $\rightarrow$ predictions $\hat{y}_i = \left(\hat{y}_i^c\right)_{c \in \mathcal{C}}$ if classification task

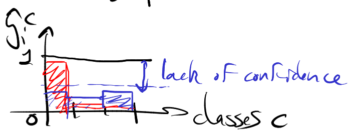
Local methods

$\hookrightarrow$ quantify the impact of the choice on the chosen sample only

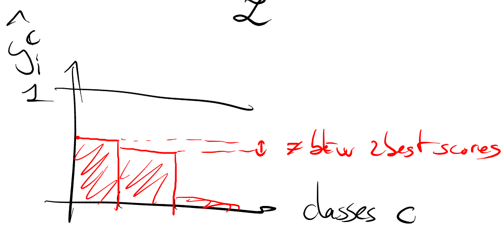
* uncertainty sampling:

pick x_i for which the model is the most uncertain $\rightarrow$ lowest prediction confidence

$$\underset{i \in \mathcal{F} \setminus \mathcal{D}_\mathcal{L}}{\arg\min} \; \sup_{c \in \mathcal{C}} \hat{y}_i^c$$



* margin sampling



$$\underset{i \in \mathcal{F} \setminus \mathcal{D}_\mathcal{L}}{\arg\min} \; \hat{y}_i^{c_1} - \hat{y}_i^{c_2}$$

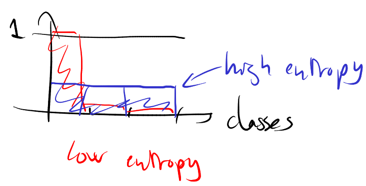
where c_1, c_2 are
$$\begin{cases} c_1 = \arg\max_c \hat{y}_i^c \\ c_2 = \text{- - - -}_{c_2 \neq c_1} \end{cases}$$

* entropy sampling:

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

entropy
proba. distrib. over classes

high: if proba. are well distributed over classes
lower: if Dirac peak



* query by committee

$\hookrightarrow$ predictor = ensemble of K models $m_k \longrightarrow \hat{y}_{i,k}$

$\hookrightarrow$ do models agree?
$\hookrightarrow$ pick x_i for which they most disagree

Global methods

$\hookrightarrow$ quantify impact of the choice over all dataset samples

* Expected model change

$\rightarrow$ do one gradient-descent step with chosen sample

Dirac peak on c^*

parameters $\theta_t \longrightarrow \theta_{t+1} = \theta_t - \eta \nabla_\theta \text{Loss}(\hat{y}_i^{c^*}, \delta_{c^*})$ $\longrightarrow \mathbb{E}_{c \sim \hat{y}_i} \; (x_i, c)$

if one knew the class for sample i

use predictions as class proba estimate

↳ expected parameter change : $\sum_{c \in \mathcal{C}} \underbrace{\hat{y}_i^c}_{\text{proba weight}} \left\| \nabla_\theta \text{Loss}(\hat{y}_i^c, \delta_c) \right\|$

↳ pick i leading to highest variation

* <u>Expected error/reduction reduction</u>

$$\underset{i}{\text{argmin}} \underbrace{\sum_{c \in \mathcal{C}} \hat{y}_i^c}_{\text{expectations over possible labels for } x_i} \sum_j \text{prediction error for } x_j \text{ if trained with } (x_i, \delta_c) \text{ also}$$

First-order dvp

$$\hookrightarrow \hat{y}_j = f_{\theta_{t+1}}(x_j) \simeq f_{\theta_t}(x_j) + \underbrace{(\delta\theta)}_{\theta_{t+1}-\theta_t} \cdot \underbrace{\nabla_\theta f_\theta(x_j) + O(\delta\theta^2)}$$

$$\underbrace{-\eta \underset{d \sim \hat{y}_i}{\mathbb{E}} \left[\nabla_\theta \text{Loss}(x_i, \delta_c) \right]}$$

$\nabla_\theta f_\theta(x_i) \cdot \nabla_\theta f_\theta(x_j)$

$\uparrow$ sample chosen for active learning — $\uparrow$ any sample from dataset

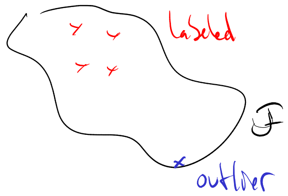
choose x_i similar to many samples x_j ⟵ similarity between samples ⟵ similarity between gradients ⟵

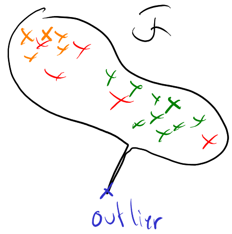
* <u>density-weighted methods</u>

↳ search for samples representative of many other ones

$$\underset{i}{\text{argmax}} \left(\underset{\times \text{ vs } \times}{\text{information brought by } (x_i, y_i)} \right) \times \sum_j \underset{\times \text{ vs } \times}{\text{similarity}(x_i, x_j)}$$

→ why?





III Incorporation of priors

- small data
- help the training of the network by adding priors from physical knowledge

A) Invariance

Enforcement of invariance by design

- symmetry of the problem : group of transformations G

$$\forall g \in G, \quad f(x) = f(gx)$$

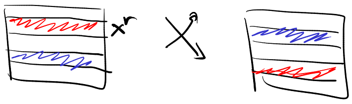
→ no need to learn it
→ easier training $\simeq$ data augmentation

- translation equi-variance: conv nets

$$F(\underset{\tau}{\text{Translated}}(x)) = \underset{\tau}{\text{Translated}}(F(x))$$

- permutation invariance:

one input: x



row permutation

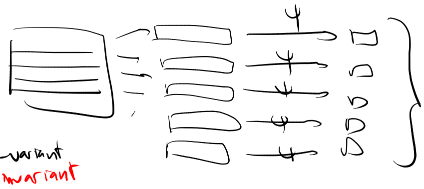
[Deep Sets]

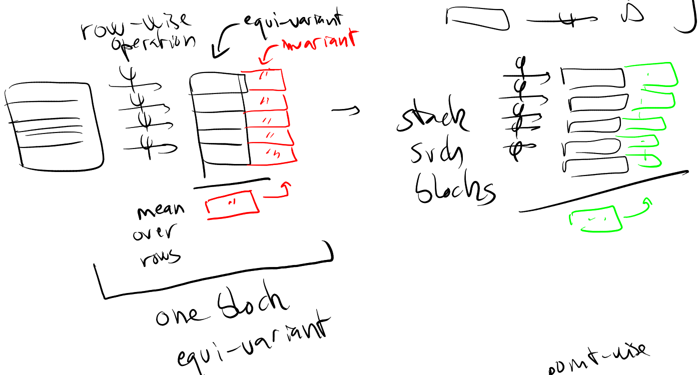
Theorem: any permutation-invariant function can be re-written as.

$$F(x) = g\left(\sum_r \psi(x^r)\right)$$

$\forall F \ldots$

$\exists \psi, g \ldots$





Th: universal permut°-equivariant approximator

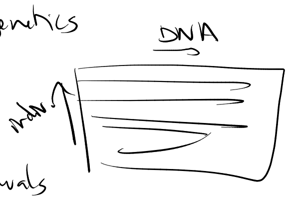
eg: point clouds → pointNet++

point-wise laser measurements on an object → 3D object classification

one input = a set of points

↳ point order-invariant

eg: population genetics



input: DNA from d individuals living now

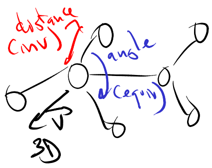
indiv. order-invariant

Task: demography size inference

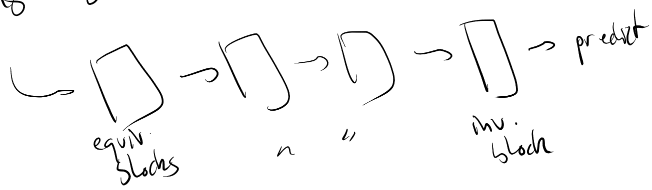
↳ how many people were living 5000 years ago?

- {invariance / equivariance} to rotations

input: molecule



→ predict toxicity

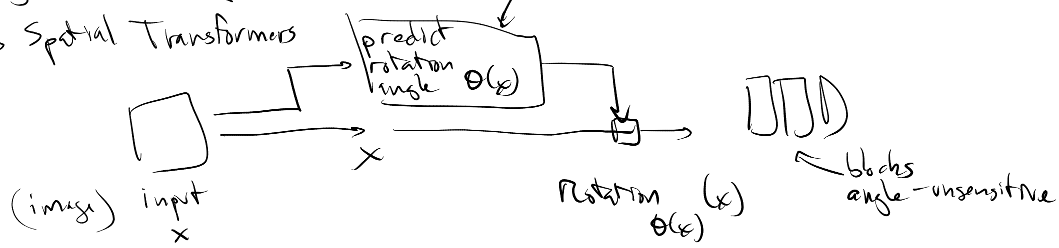


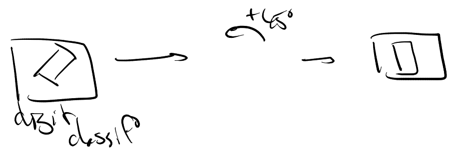
$$v \to \|v\| \text{ norm} \xrightarrow{\psi} \cdots \quad\quad \hookrightarrow \psi(\|v\|)\,\frac{v}{\|v\|}$$

$$\to \text{direction } \frac{v}{\|v\|}$$

Learning the invariance

→ Spatial Transformers





→ Capsule networks
not inv : equiv

B) by task design → metrics
→ data