Stanford University

CS 329P : Practical Machine Learning (2021 Fall)

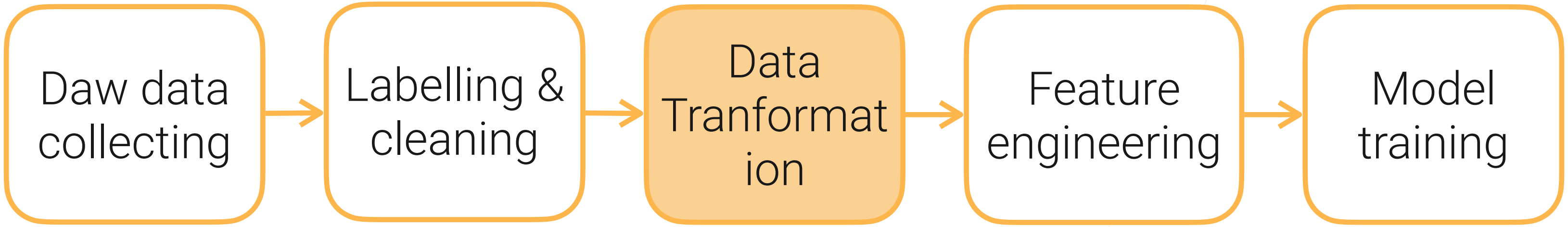
# 2.3 Data Transformation

Qingqing Huang, Mu Li, Alex Smola

https://c.d2l.ai/stanford-cs329p

# Data Transformation

- ML algorithms prefer well defined fixed length, well-conditioned, nicely distributed input
- Next, data transformation methods for different data types



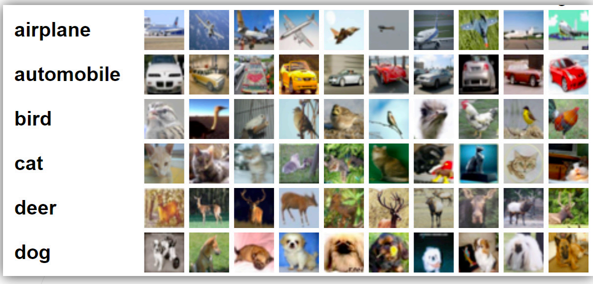
# Normalization for Real Value Columns

- Normalization makes training more stable

| | |
|---|---|
| Min-max normalization: linearly map to a new min *a* and max *b* | $x_i' = \frac{x_i - \text{min}_{\mathbf{x}}}{\text{max}_{\mathbf{x}} - \text{min}_{\mathbf{x}}}(b - a) + a$ |
| Z-score normalization: *0* mean, *1* standard deviation | $x_i' = \frac{x_i - \text{mean}(\mathbf{x})}{\text{std}(\mathbf{x})}$ |
| Decimal scaling | $x_i' = x_i/10^j$ smallest $j$ s.t. $\text{max}(\lvert\mathbf{x}'\rvert) < 1$ |
| Log scaling | $x_i' = \log(x_i)$ |

# Image Transformations

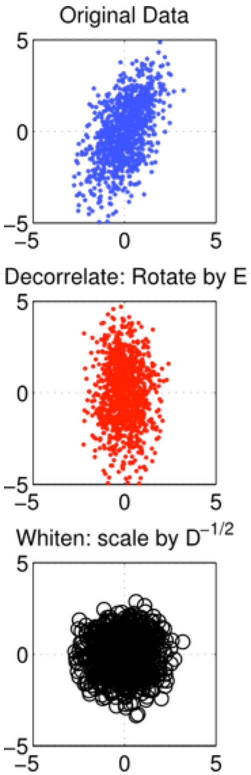
- Our previous web scraping will scrape 15 TB images for a year
    - 5 millions houses sold in US per year, ~20 images/house, ~153KB per image, ~1041x732 resolution
- cropping, downsampling, compression
    - Save storage cost, faster loading at training
        - At ~320x224 resolution, 15 TB -> 1.4TB
    - ML is good at low-resolution images
    - Be aware of lossy compression
        - Medium (80%-90%) jpeg compression may lead to 1% acc drop in ImageNet



CIFAR-1

# Image Transformations

- Image whitening
  - Generalized normalization of vector values
  - Pixels in local neighborhood are highly correlated.
  - Whitening removes redundancy through linear transformations
    - Vector $x$ has mean 0 and covariance estimate $\Sigma$
    - $y = Wx$ , st $W^T W = \Sigma^{-1}$. $y$ has unit diagonal covariance
    - Common choices of whitening matrix: Eigen-system of $\Sigma$(PCA), $\Sigma^{-\frac{1}{2}}$ (ZCA),
  - Model converges faster with whitened image input
    - Especially for unsupervised learning, e.g. GAN


Original Data

Decorrelate: Rotate by E

Whiten: scale by $D^{-1/2}$

PCA whitening

# Video Transformations

- Input variability high
  - Average video length: Movies ~2h, YouTube videos ~11min, Tiktok short videos ~15sec
- Tractable ML problems with short video clips (<10sec)
  - Ideally each clip is a coherent event (e.g. a human action)
  - Semantic segmentation is extremely hard..
- Preprocessing to tradeoff storage, quality and loading speed
- Common practice: decode a playable video clip, sample a sequence of frames, compute spectrograms for audio
  - Easy to load to model, increased storage space

# Text Transformations

- Stemming and lemmatization: a word → a common base form
  - E.g. am, are, is → be     car, cars, car's, cars' → car
  - Example: Topic modeling
- Tokenization: text string→ a list of tokens (smallest unit to ML algorithms)
  - By word: `text.split(' ')`
  - By char: `text.split(‘')`
  - By subwords:
    - e.g. “a new gpu!” → “a”, “new”, “gp”, “##u”, “!”
    - Custom vocabulary learned from the text corpus (Unigram, WordPiece)

# Summary

- Transform data into formats preferred by ML algorithms
  - Tabular: normalize real value features
  - Images: cropping, downsampling, whitening
  - Videos: clipping, sampling frames
  - Text: stemming, lemmatization, tokenization
- Need to balance storage, quality, and loading speed