Few Shot Learning

The Israeli Statistical Association Workshop

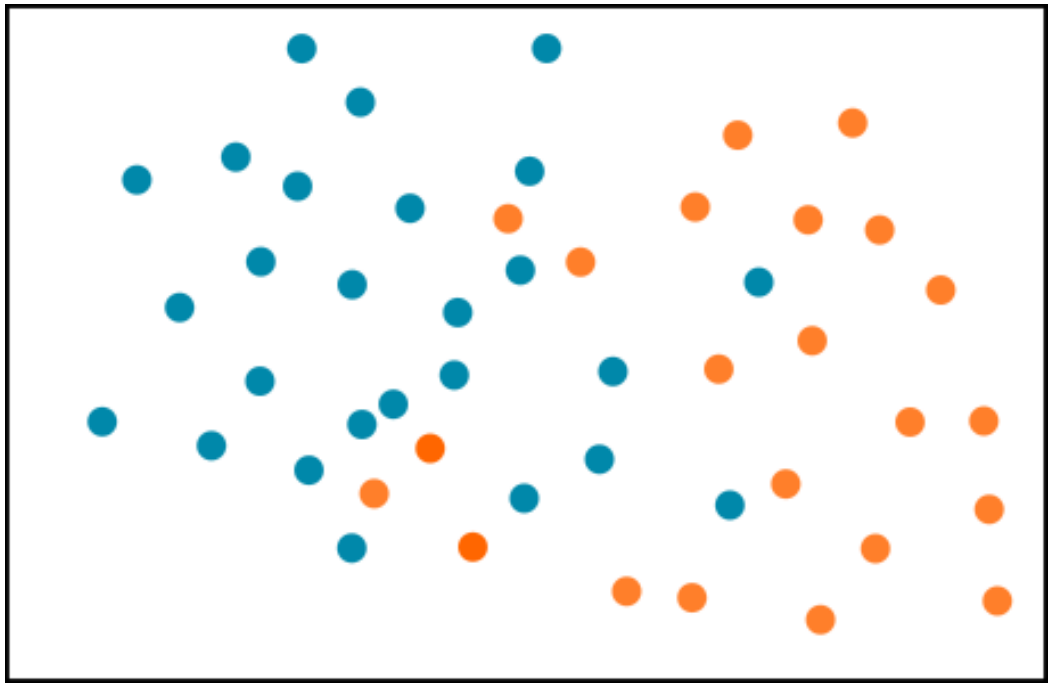
Yuli Slavutsky

December 2022

Introduction

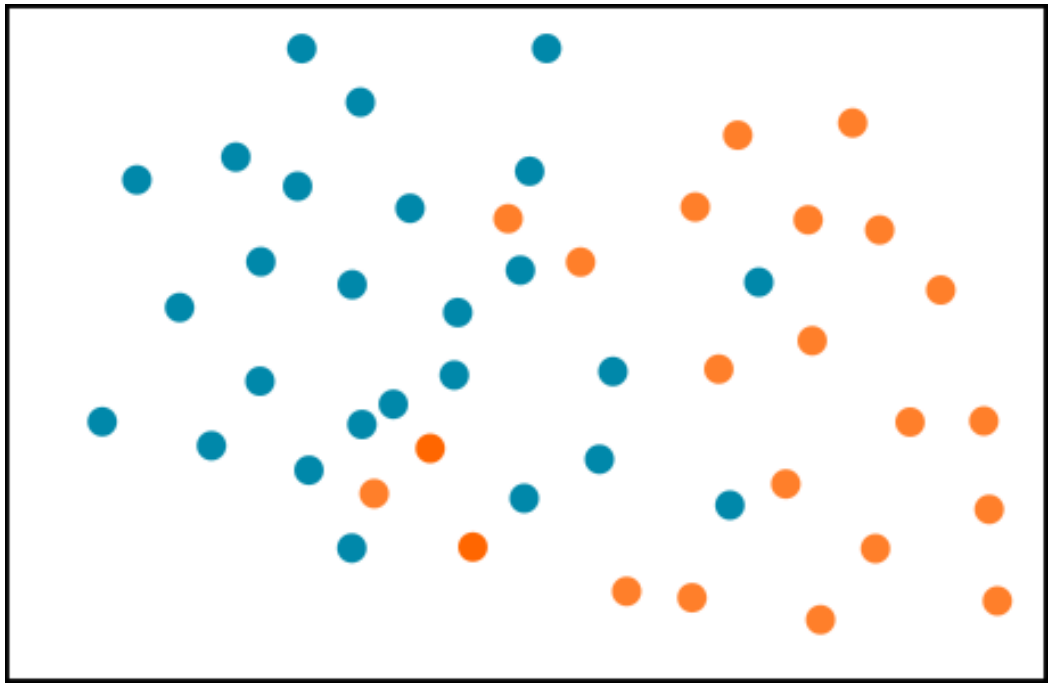
Standard supervised learning

Suppose we have a relatively simple classification task



Standard supervised learning

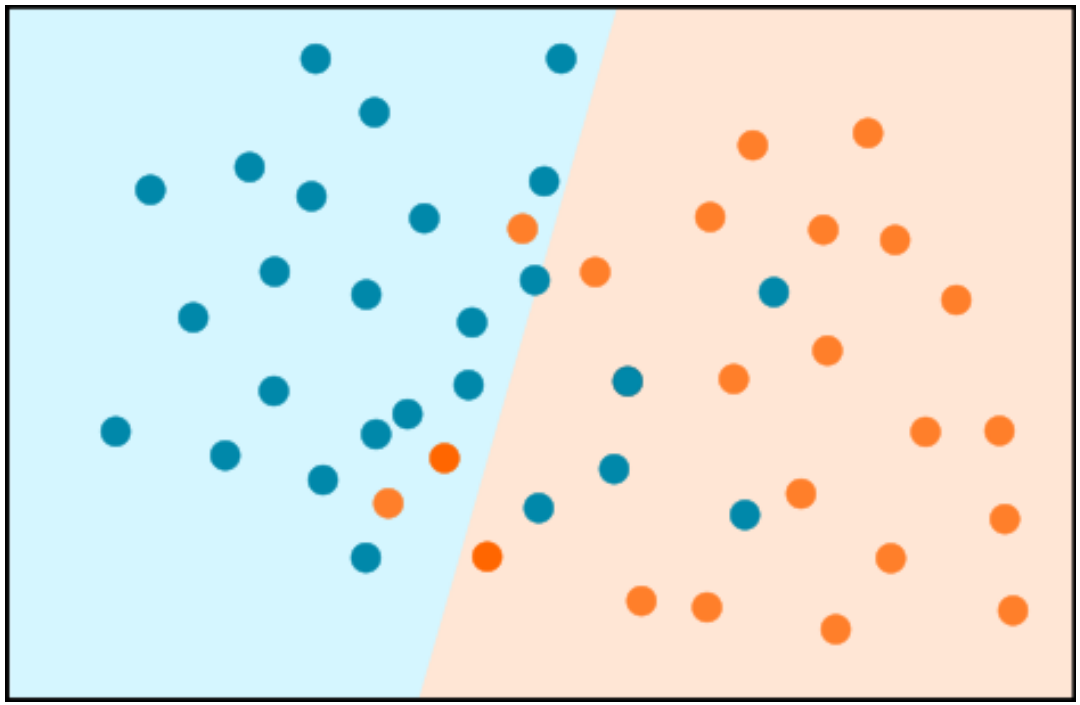
A classical statistical approach: assign a model $P(x|y=k)$, $k \in \{1,2\}$



Standard supervised learning

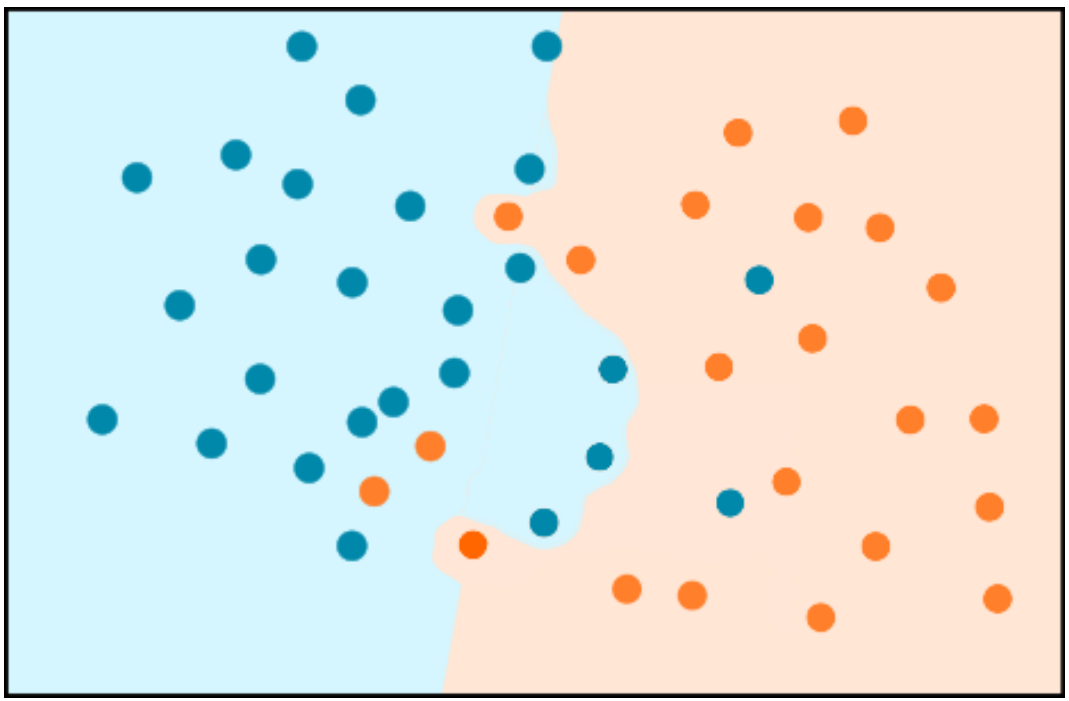
For example, by modeling $P(x|y=k) = \mathcal{N}(\mu_k, \Sigma_k)$ we get Linear Discriminant Analysis:

$$P(y=k|x) = \frac{P(x|y=k)}{P(x)} = \frac{P(x|y=k)}{\sum_l P(x|y=l)P(y=l)}$$



Standard supervised learning

Some other techniques such as the K-nearest neighbors algorithm do not require directly modeling $P(x|y = k)$, and instead rely on the distances between data-points:



Standard supervised learning

What did we assume so far?

Assumptions

Common assumptions:

- The data is relatively low dimensional, and easiy to model/assign distances for
- The set of possible labels $\mathcal{Y}$ is small
- The training data $T = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ includes large amounts of instances $x \in \mathcal{X}$ for each class

Today we will violate all these assumptions

Assumptions

Data is not always simple enough to apply modeling/distance based techniques directly

Image source: https://blogs.microsoft.com/on-the-issues/2018/12/06/facial-recognition-its-time-for-action/

Assumptions

- We would like to project the data to another space
  - Should we ignore glasses?
  - Hair color? Eye color?
  - Positions of which face parts are important?
- This is the task of representation learning
- Seek a function g such that $g(x) = \hat{x} \in \hat{\mathcal{X}}$
- The classification is easier in $\hat{\mathcal{X}}$

Assumptions

- The face recognition task is different from standard supervised learning in another aspect
- Think about the face-recognition algorithm in your smart phones
- It was probably trained on images of a lot of other people
- When you first bought it, it asked you take a few pictures of you
- Did it perform re-training?
- How (and why) it works?

Examples

Forensic science (fingerprint identification)

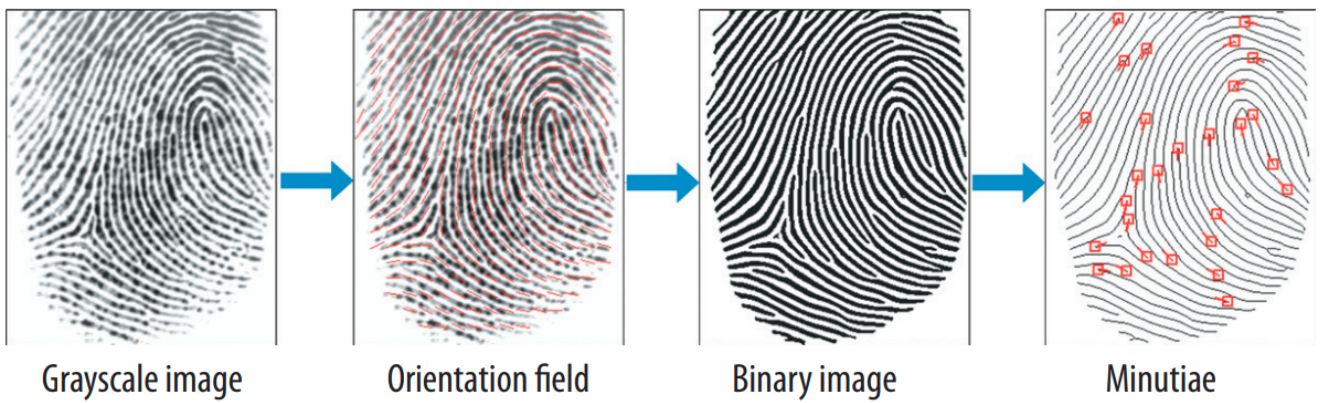


Image source: Jain, A. K., Feng, J., and Nandakumar, K. (2010). Fingerprint matching. Computer, 43(2), 36-44.

Examples

Speaker recognition

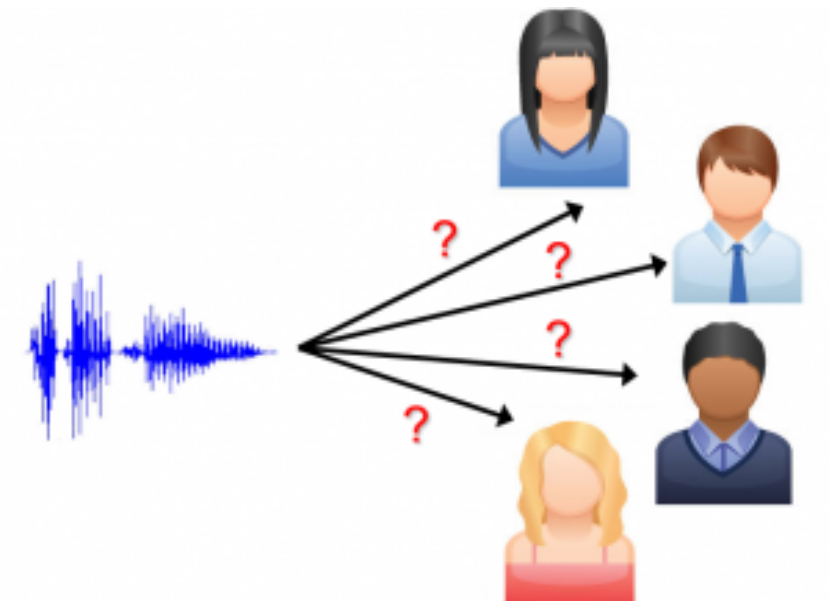
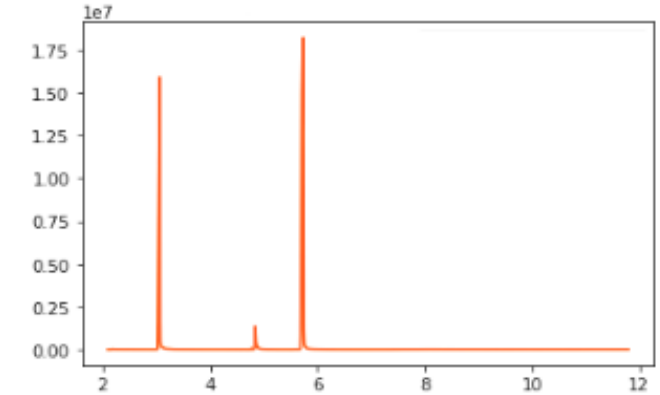
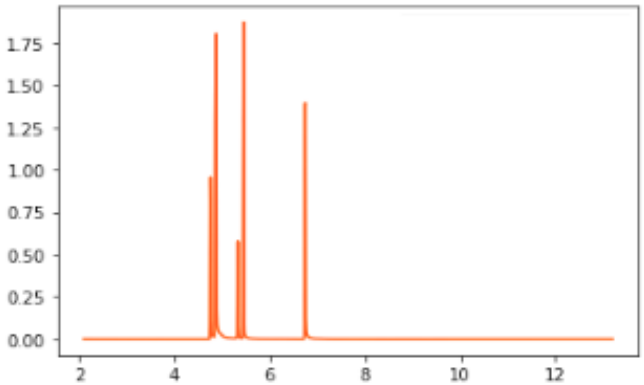


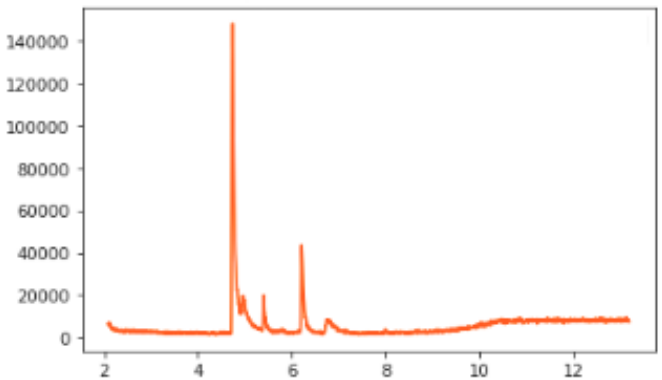
Image source: https://partner.phonexia.com/kb/sp/speech-platform/spe/technologies-available-spe/speaker-identification-sid/

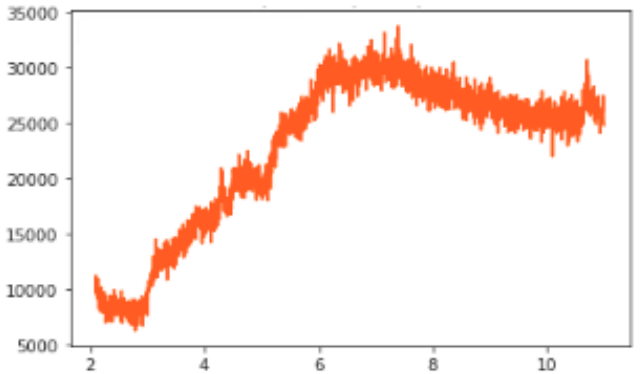
Examples

Drug mixture recognition









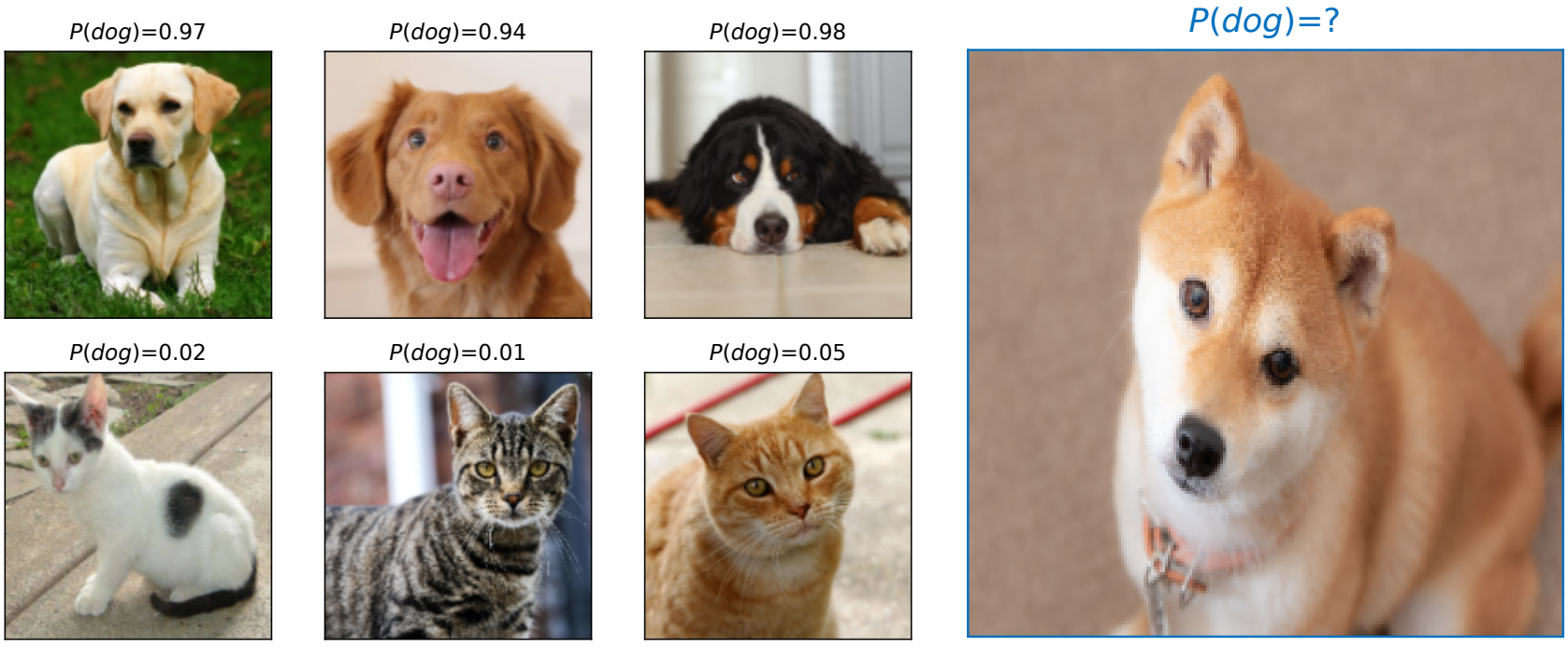
What is few-shot learning?

Few-shot learning

- Originally introduces by Fei-Fei et al. (2006)
- In few-shot learning we violate the assumption that there are ”enough” training instances from each class
- Think of ”few” = 1
- We often violate the assumption that the label space is small

Generalization in standard supervised learning

In standard settings we consider the case of a new data point x, from one of the classes seen in the training set T



Generalization in few-shot learning

Few shot learning introduces another challange: new classes

Data

- N-way K-shot: N new classes, with K labeled examples each
- T - Training set, includes large amounts of instances from each "old" class $y \in \mathcal{Y}_{\text{old}}$
- S - Support set, consists of the K labeled samples from the N "new" classes $y \in \mathcal{Y}_{\text{new}}$
- Q - Query set, consists of instances from "old" and "new" categories, the data on which the model needs to generalize

Data

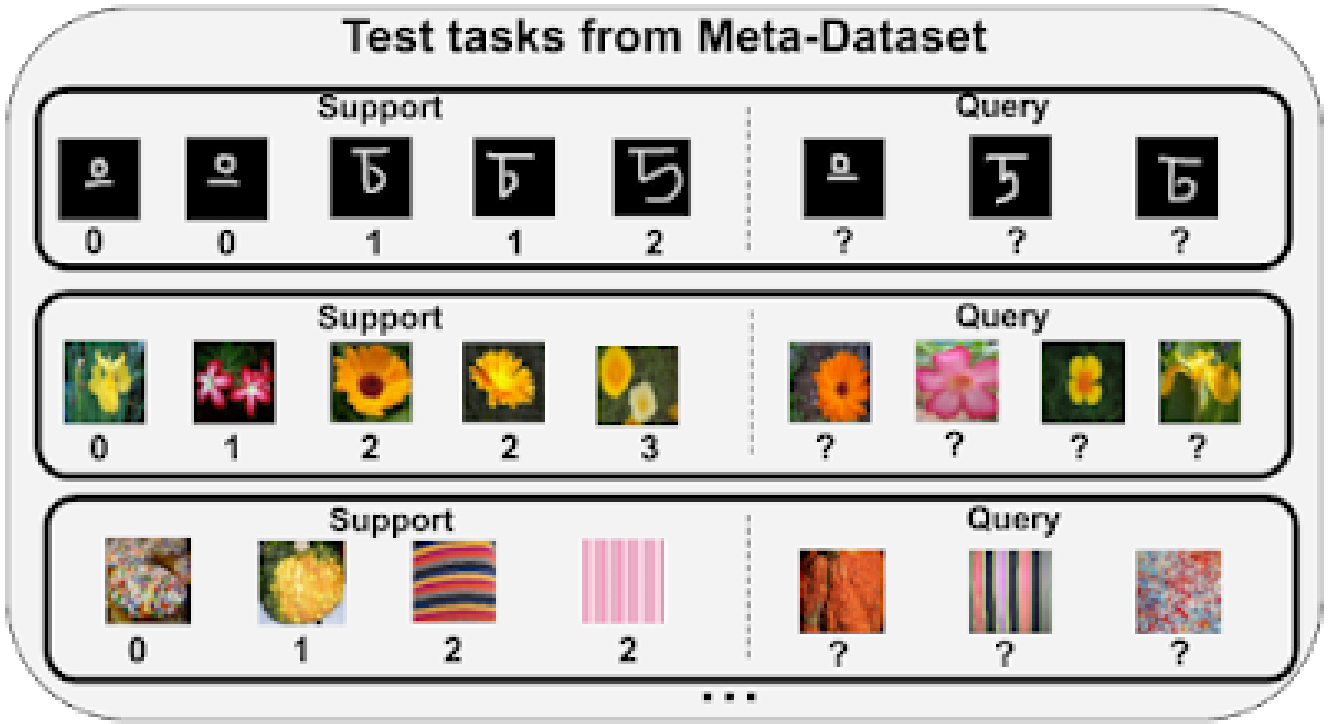


Image source: https://ai.googleblog.com/2020/05/announcing-meta-dataset-dataset-of.html

Few-shot learning

The hard part:

Learn to classify instances from N new classes with only K examples

Zero-shot learning

- A special case is zero-shot learning
- The learner is required to discriminate between N new classes, with no training examples from them

Few-shot learning

The main idea:

Take advantage of knowledge coming from previously learned tasks

Transfer-Learning

Transfer-Learning

- Transfering ”knowledge” gained on one task, for solving another
- Often applied in the context of neural networks
- We can view the neural-network based classifier as a function composition $h(g_\theta(x))$ where we treat the first m layers as the representation $g_\theta(x) = \hat{x}$, and the rest as the classifier.

Transfer-Learning

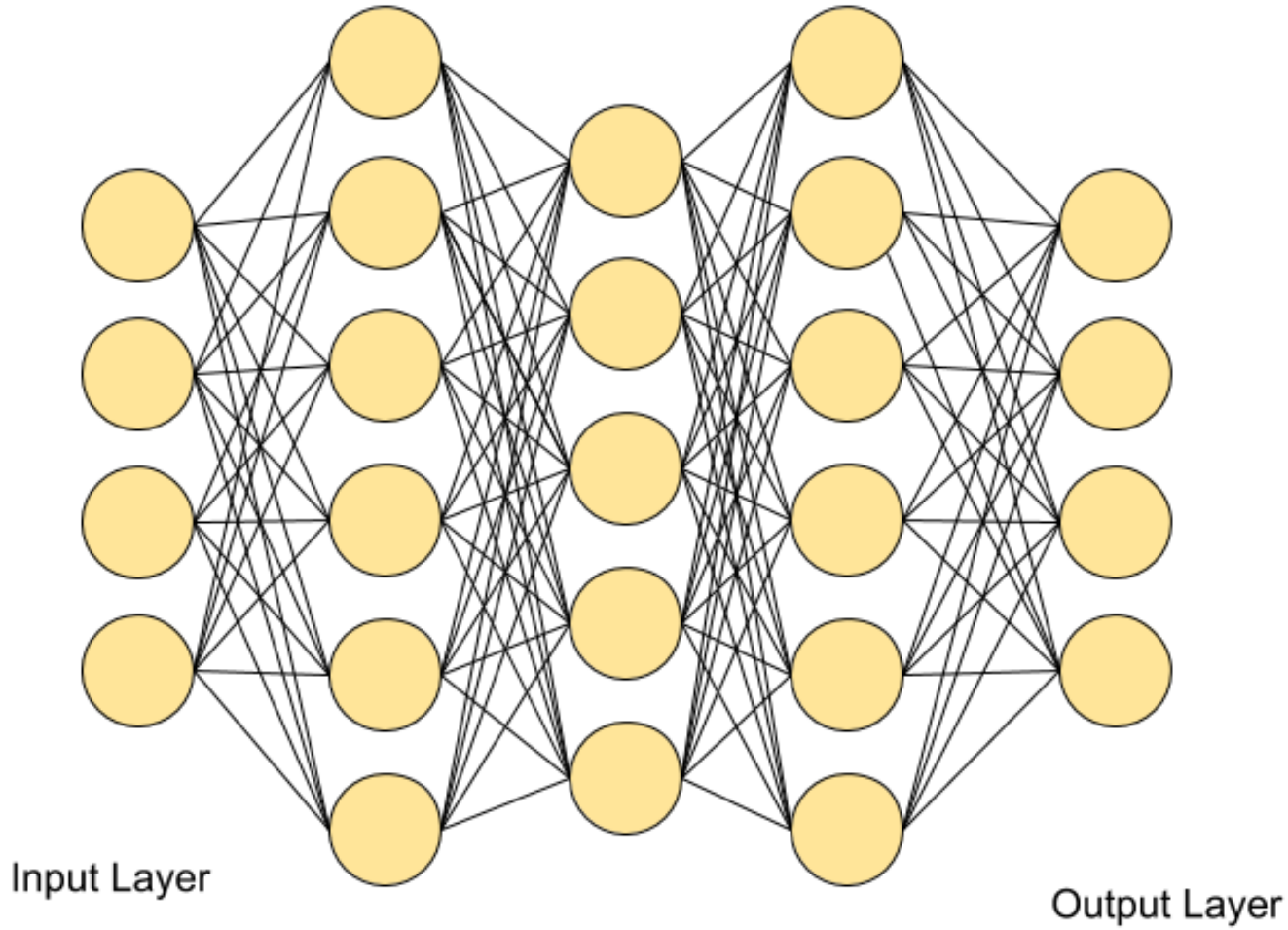


Image source: https://www.danrose.ai/blog/transfer-learning-from-a-business-perspective

Transfer-Learning

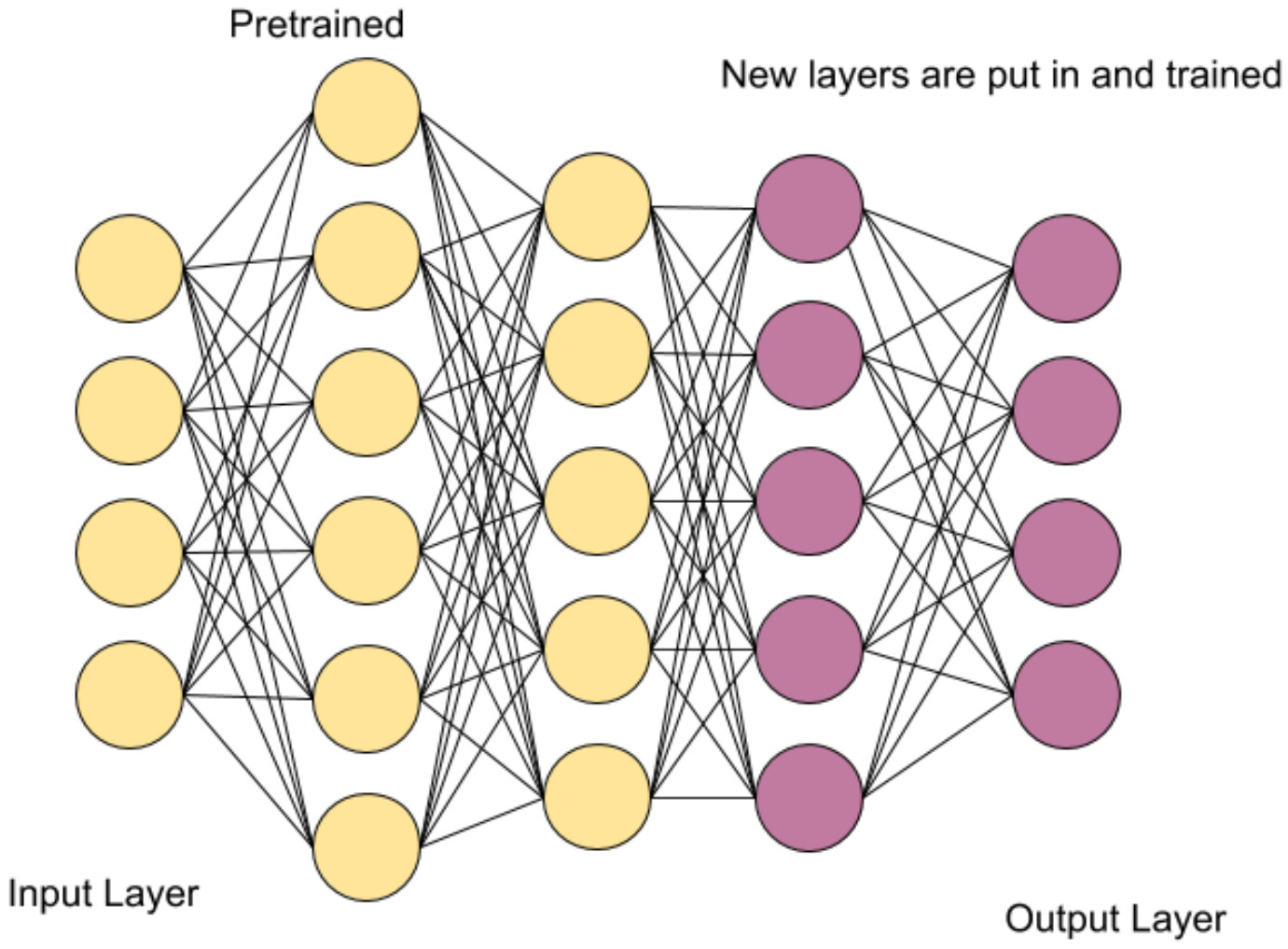


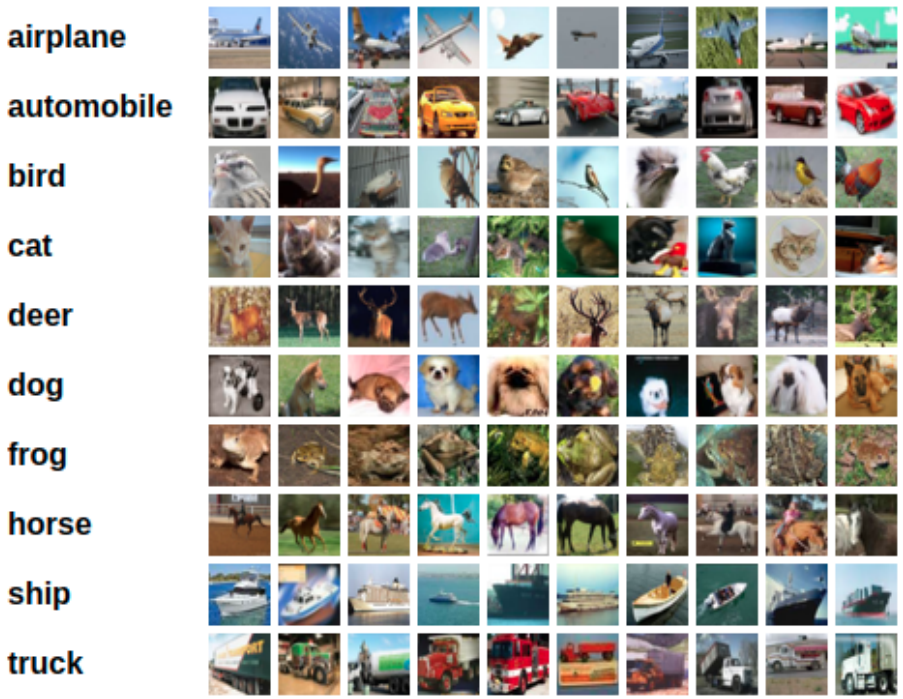
Image source: https://www.danrose.ai/blog/transfer-learning-from-a-business-perspective

Transfer-Learning

Assume we have a neural-network trained on Imagenet

Transfer-Learning

We can leverage the learned representation for similar object recognition tasks



CIFAR-10 dataset

Meta-Learning

Connection to Meta-learning

- Few-shot learning can be viewed a type of Meta-Learning
- Assume a distribution over classification tasks τ
- Each task is N-way K-shot, has its own support and query
- The idea: Train on similar tasks that contain different classes
- The process: At each step update the model parameters based on a randomly selected training task

Meta-learning approaches

1. Prior knowledge about the structure and variability of the data
   - Learning from data augmentations
     (Hariharan & Girshick, 2017)
   - Generative models
     (Lake et al., 2015; Edwards & Storkey, 2016)
2. Prior knowledge about the learning process
   - Learn initial parameters that can be easily fine-tuned (closely related to transfer-learning):
     - MAML and FOMAMl (Finn et al., 2018),
     - Reptile (Nichol & Schulman, 2018)
   - Update rules suitable for small datasets
     (Ravi & Larochelle, 2017; Li & Malik, 2017; Bello et al., 2017)

Meta-learning approaches

3. Prior knowledge about similarities: learn representations that facilitate discrimination between classes

   - Pairwise comparisons:
     Triplet Networks (Hoffer & Ailon, 2015)
     Siamese Networks (Koch et al., 2015)

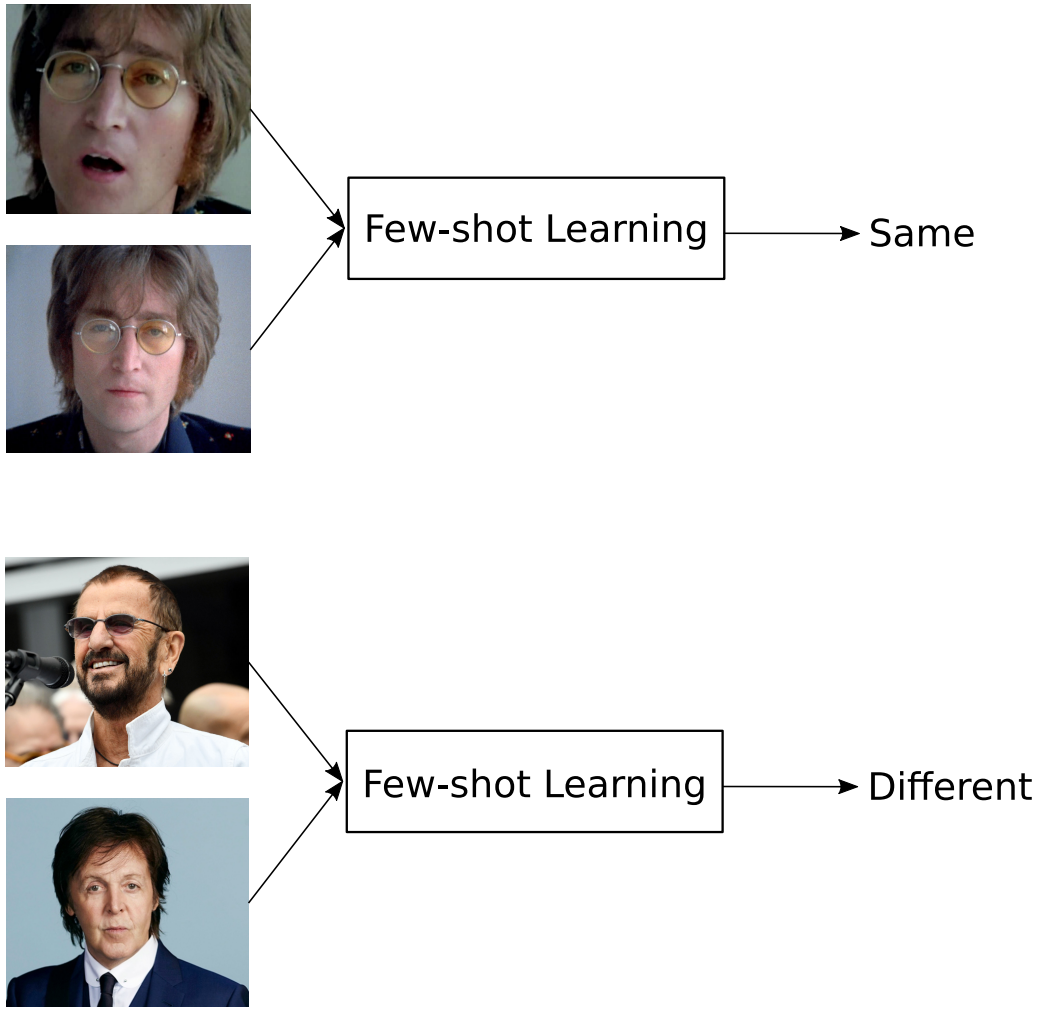
This will be our focus today

Beyond class-specific models

Moving beyond class-specific models

- In standard classification tasks the learner outputs either the correct class or a score for each previously seen class
- What about new classes?
- We now require a class-agnostic solution

Moving beyond class-specific models



Moving beyond class-specific models

- Binary classification: given a pair, determine if they are from the same class
- Our new focus: learn data representations on which we can apply similarities
  - We want instances from the same class to be close
  - and instances from different classes to be far
- Then we can apply (essentially) k-nearest neighboor techniques

Distance focused representation learning

Distance focused representations

- Learning good data representations is one of the leading challenges of deep learning
- For applying distance-based techniques we need to choose a distance metric
- Generally there is no guarantee that a representation learned for some task will generate meaningful distances in the metric we chose
- Therefore we will train it according to a pre-chosen distance metric explicitly

Benefits of distance based training of representations

1. Simple classification rules (threshold on distance)
2. Intuitive representation spaces
3. Allows scientific analysis of distances between representations (e.g in neuroscience)

Distance focused representation learning

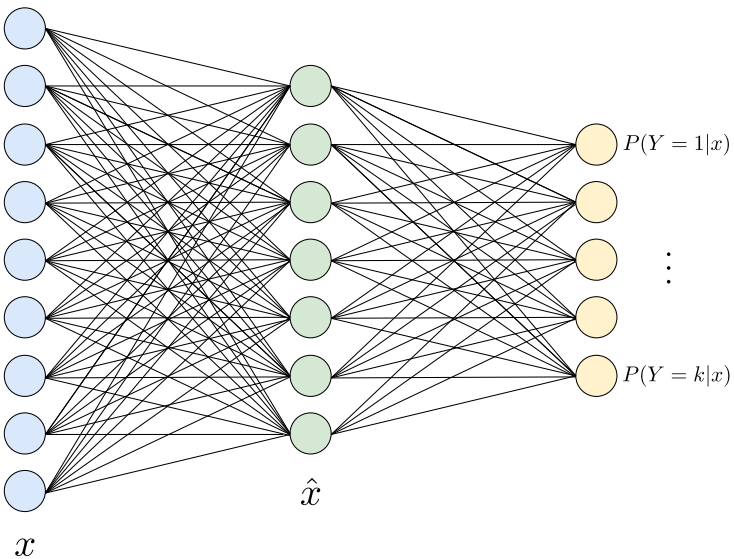
Siamese neural networks

Siamese neural networks

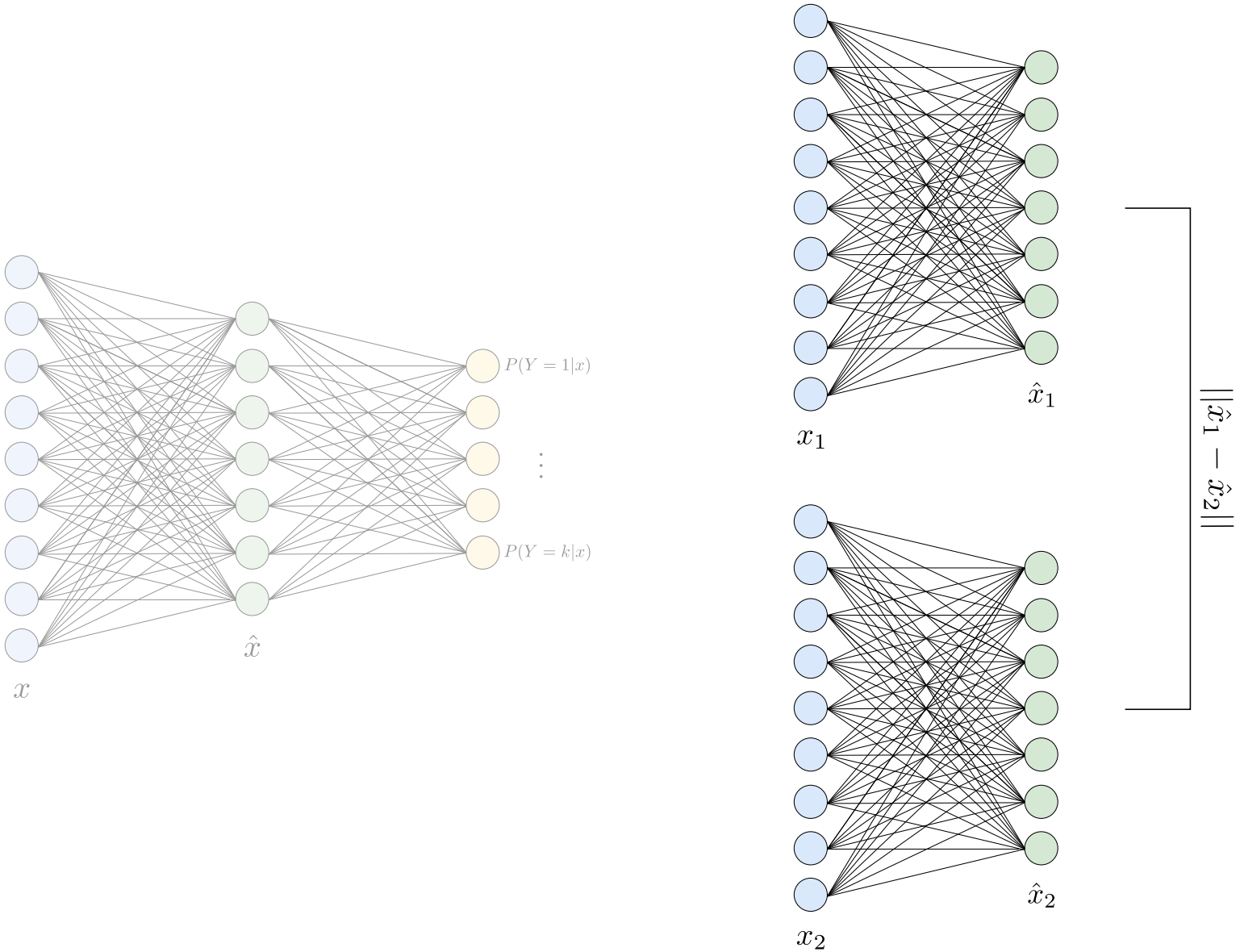
- Assume that the output of the classifier is $h(g_\theta(x))$
  - g_θ is the representation
  - $h(g_\theta(x)) \in [0, 1]$ is a score for a pair to belong to the same class
- Two examples are passed through identical networks (hence Siamese) to create two representations
- $d_{ij} = \left\| g(x_i) - g(x_j) \right\|$
- A binary loss is applied

$$\ell(x_i, x_j) = -\left[\mathbb{1}_{\{y_i = y_j\}} \log \sigma(d_{ij}) + \left(1 - \mathbb{1}_{\{y_i = y_j\}} \log \left(1 - \sigma(d_{ij}) \right) \right) \right]$$

Siamese neural networks



Siamese neural networks



Siamese neural networks

- During training, each pair of examples are randomly drawn
- The learner learns to discriminate between classes is general, rather than two classes in particular
- At test time completely different classes are used

Distance focused representation learning

Triplet Networks

Triplet Loss

Denote $\hat{x} = g_\theta(x)$.

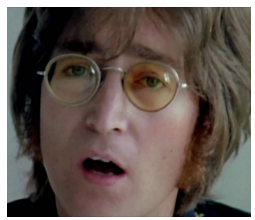
Triplet Loss:

x_a - anchor

x_p - a positive sample (from the same class as x_a)

x_n - a negative sample (from a different class as x_a)

x_a



x_p

x_n

Optimize g_θ such that $\left\|\hat{X}_a - \hat{X}_p\right\|^2$ is small and $\left\|\hat{X}_a - \hat{X}_n\right\|^2$ large.

Triplet loss

Optimizing $\left\|\hat{X}_a - \hat{X}_p\right\|^2 - \left\|\hat{X}_a - \hat{X}_n\right\|^2$ may result in a trivial collapse.

We introduce a margin α:

$$\ell(x_a, x_p, x_n; \theta) = \max\left(\left\|\hat{X}_a - \hat{X}_p\right\|^2 - \left\|\hat{X}_a - \hat{X}_n\right\|^2 + \alpha, 0\right)$$

And the risk function is:

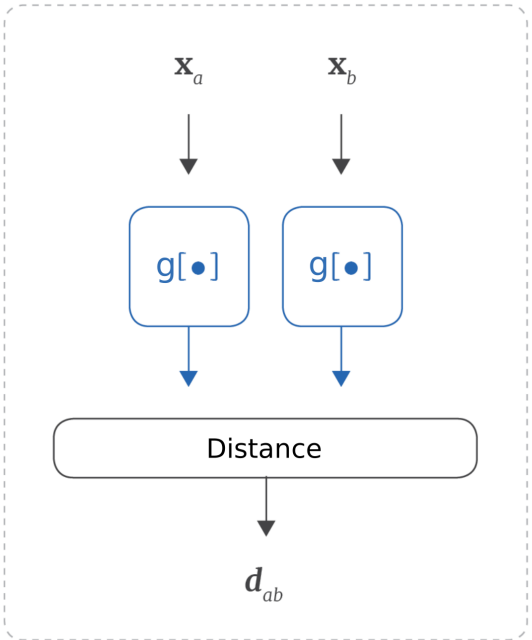
$$L_T = \frac{1}{N}\sum_{i=1}^{N} \ell\left(X_a^{(i)}, X_p^{(i)}, X_n^{(i)}; \theta\right)$$

Triplet networks

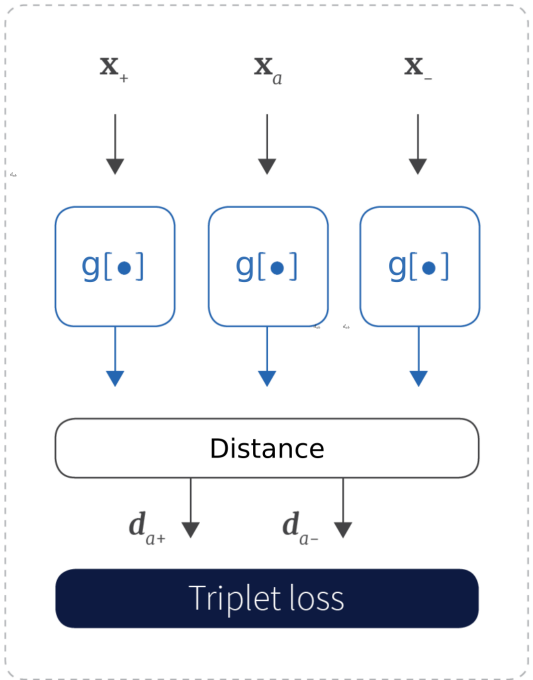
- During training, anchors are drawn, and for them positive and negative examples
- At test time the learner can establish whether pairs are from the same class by thresholding the distance

Pairwise comparisons

a) Siamese network



b) Triplet network (training)



c) Triplet network (testing)

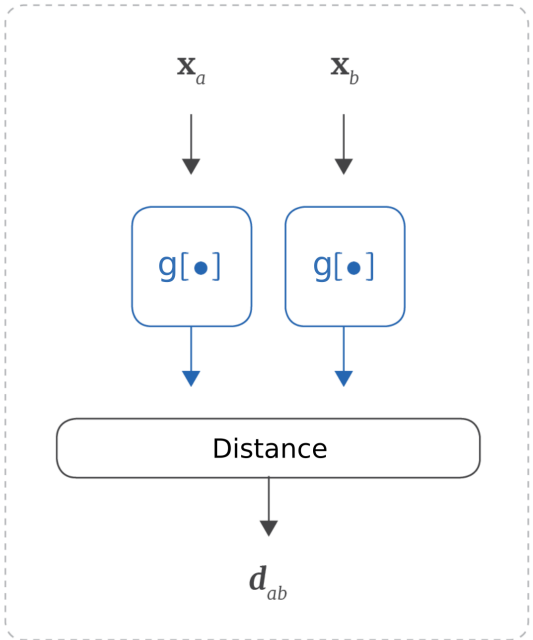


Image adapted from https://www.borealisai.com/research-blogs/tutorial-2-few-shot-learning-and-meta-learning-i/

Contrastive learning

Similarly to triplet loss, contrastive loss (Hadsell et al., 2006) is defined as:

$$\ell(x_i, x_j; \theta) = (1 - \mathbb{1}_{\{y_i = y_j\}}) \frac{1}{2} \left\| \hat{X}_i - \hat{X}_j \right\|^2 + \mathbb{1}_{\{y_i = y_j\}} \frac{1}{2} \left[\max \left(\alpha - \left\| \hat{X}_i - \hat{X}_j \right\|, 0 \right) \right]^2$$

And the cost function is:

$$L_T = \sum_{i,j} \ell \left(x_i, x_j; g \right)$$

Note that although very similar to triplet-networks, contrastive loss is applied on pairs.

Hands-on session

Hands-on session

https://bit.ly/FewShot

Slides: *https://github.com/YuliSl/Few-Shot-Tutorial*