Deep Learning

Lecture 10: Meta and manifold learning

Chris G. Willcocks

Durham University

Lecture overview

1. **Manifold learning**
   - NLDR with DNNs
   - t-SNE and UMAP on DNNs
   - designing tailored embeddings
   - Jonker-Volgenant assignment
2. **Meta learning**
   - thinking in distributions
   - the distribution of all data...
   - ...and of all tasks
   - definition
   - the meta learning support set
   - metric, optimisation and model-based
3. **Looking forward**
   - meta learning datasets
   - large-scale generative models
   - machine reasoning and risk
   - take away points

Definition: NLDR in DNNs

Feature vectors in deep neural networks (DNNs) capture abstract patterns which are interesting to analyse.

We can use nonlinear dimensionality reduction (NLDR) algorithm, such as t-SNE and UMAP to examine these patterns.

The deepest (bottleneck or penultimate layer) features are often the most interesting.

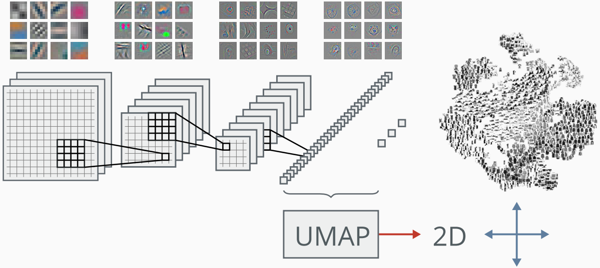
Example: bottleneck features in LeNet

```
class LeNet(nn.Module):
  def __init__(self):
    super(LeNet, self).__init__()
    self.conv1 = nn.Conv2d(1, 6, 5, padding=2)
    self.conv2 = nn.Conv2d(6, 16, 5)
    self.fc1   = nn.Linear(16 5 5 , 120)
    self.fc2   = nn.Linear(120, 84)
    self.fc3   = nn.Linear(84, 10)

  def forward(self, x):
    x = F.max_pool2d(F.relu(self.conv1(x)), (2, 2))
    x = F.max_pool2d(F.relu(self.conv2(x)), (2, 2))
    x = flatten(x)
    x = F.relu(self.fc1(x))
→   f = F.relu(self.fc2(x))
    x = self.fc3(f)
    return x
```

Manifold learning example t-SNE and UMAP on DNNs

Usage: t-SNE [1] or UMAP [2]

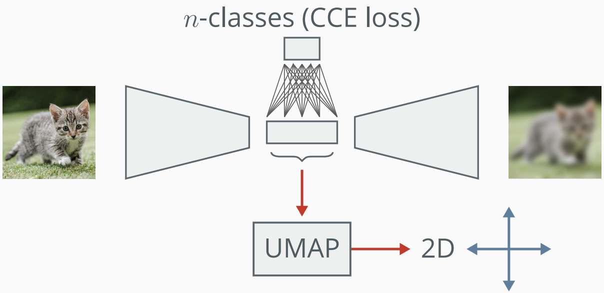
```
import torch
from sklearn.manifold import TSNE

# f = features for whole dataset
f = torch.randn(1000, 84, 1, 1)

# specify embedding to 2D
g = TSNE(2).fit_transform(f.squeeze())
print(g.shape) # returns (1000,2)
```



Example: tailored embeddings

The embedding space can be controlled by additional constraints, such as reconstruction term, additional losses (classificaiton, regression).

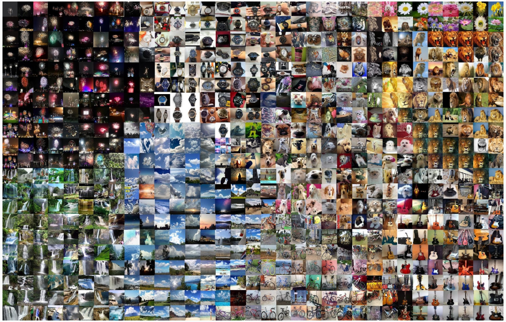
What will the 2D embedding be like for the following architecture?

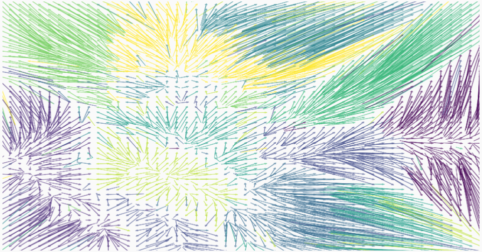


Manifold learning Jonker-Volgenant assignment

Example: Jonker-Volgenant

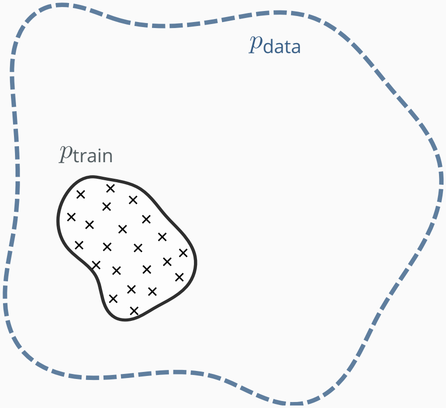
A visualisation trick is to minimise an assignment cost to optimise the layout of the embeddings. The Jonker-Volgenant algorithm can be used for this, giving:



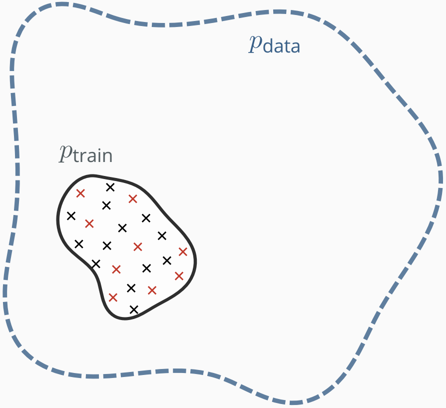


Meta learning thinking in distributions

A common lie...
...is that test data $\overset{\text{i.i.d.}}{\sim}$ train data (no!)

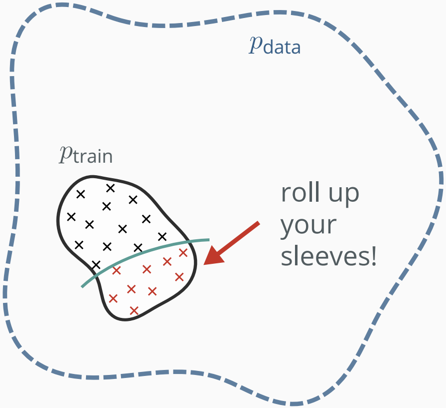


99.7% "test accuracy"!
(your boss and the investors are happy)

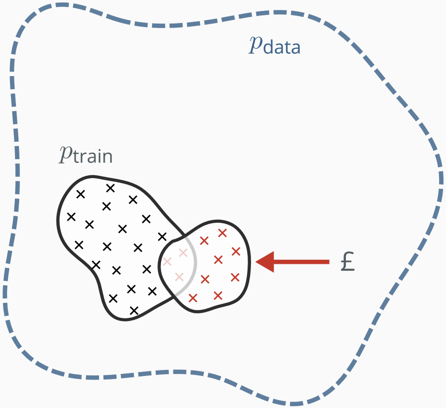


Meta learning thinking in distributions

62% "test" accuracy
(but closer to the truth)

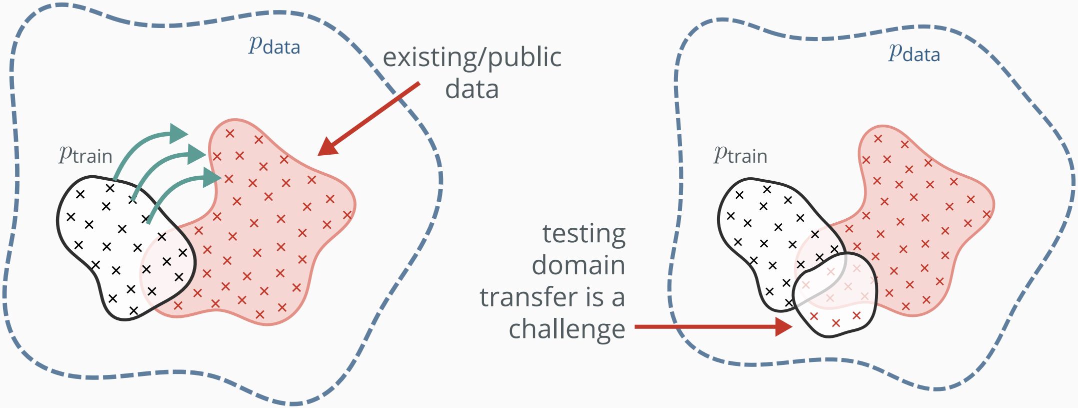


51% "test" accuracy
(even closer to the truth)

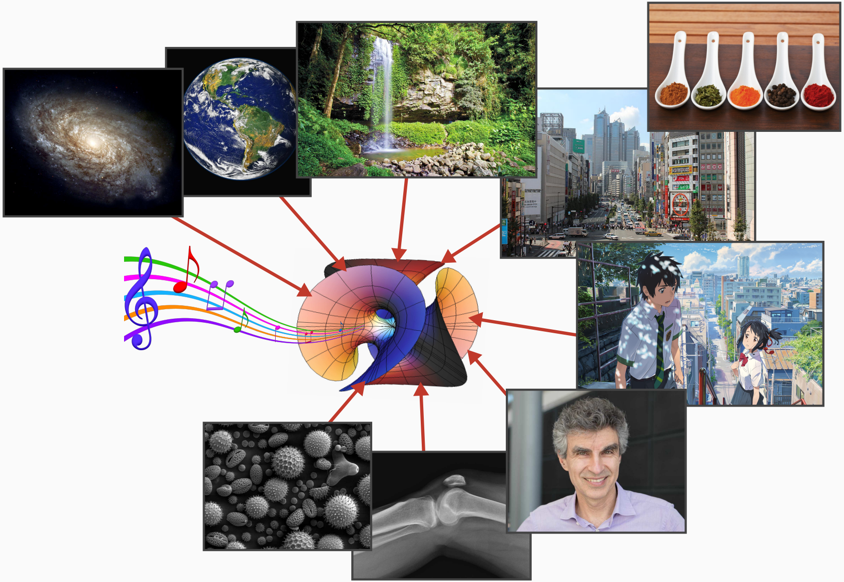


Meta learning thinking in distributions

Generative models (e.g. domain adaptation, transfer and meta learning)



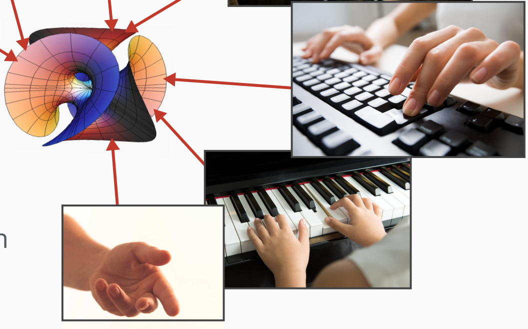
Meta learning the distribution of all data



Meta learning the distribution of all tasks

Learning to learn to:

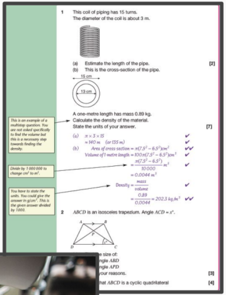
- move our fingers
- play music
- touch type
- sing, talk
- think like Einstein and Hinton
- ...



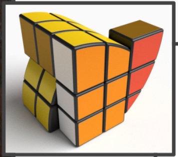
Meta learning inferring new tasks

We would like to be able to generalise to unseen tasks. What do you do with these?

Solve, comprehend...



Solve, admire...



Play, complete, improvise...

Play, measure...

Eat, taste, smell...

Meta learning definition

Definition: meta learning

Learn a distribution of (related) tasks, so we can infer new tasks quickly [3].

Instead of training on data samples $x \sim p_{\text{data}}$ we train on datasets $\mathcal{D} \sim p(\mathcal{D})$

$$\theta^* = \arg\min_\theta \mathbb{E}_{\mathcal{D}\sim p(\mathcal{D})}[\mathcal{L}_\theta(\mathcal{D})]$$

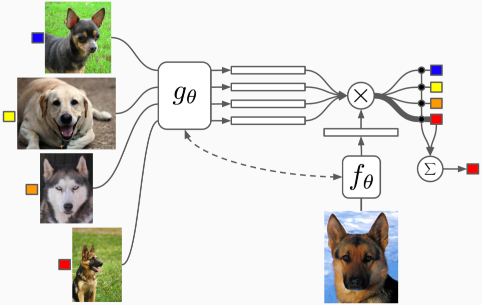
$$\theta^* = \arg\min_\theta \mathbb{E}_{\mathcal{D}\sim p(\mathcal{D})}[\mathcal{L}_\theta(\mathcal{D})]$$

Definition: meta learning support set

Meta learners determine the task via a support set $\mathcal{S}$

$$\theta = \arg\max_{\theta} \mathbb{E}_{L\sim\mathcal{T}}\left[\mathbb{E}_{\mathcal{S}^L\sim\mathcal{D}, B^L\sim\mathcal{D}}\left[\sum_{(x,y)\in B^L} P_\theta(y|x,\mathcal{S}^L)\right]\right]$$

support set S



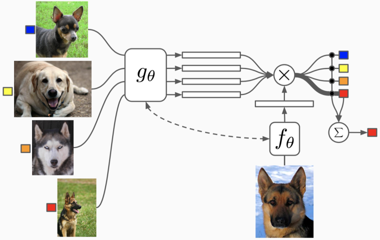
Taxonomy: meta learning

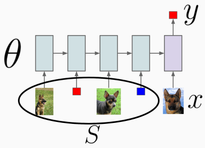
Meta learning literature can be categorised several ways [3], such as by:

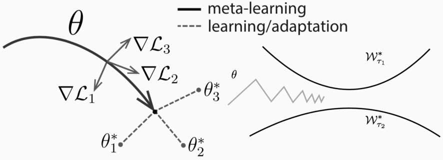
- meta-representation (what data?)
- meta-optimisation (how's it optimised?)
- meta-objective (what goal?)

or a different taxonomy:

- metric-based
  $P_\theta(y|\mathbf{x}, S) = \sum_{(\mathbf{x}_i, y_i) \in S} k_\theta(\mathbf{x}, \mathbf{x}_i) y_i$
- model-based
  $P_\theta(y|\mathbf{x}, S) = f_\theta(\mathbf{x}, S)$
- optimisation-based
  $P_\theta(y|\mathbf{x}, S) = P_{g_\phi(\theta, S^L)}(y|\mathbf{x})$





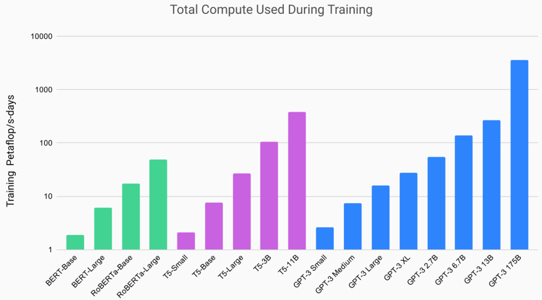


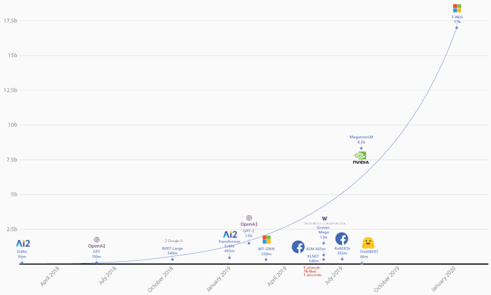
Looking forward meta learning datasets

Omniglot

Mini-Imagenet

Looking forward large-scale generative models



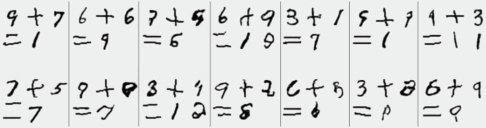


Figures from OpenAI and Hugging Face.

Meta learning machine reasoning

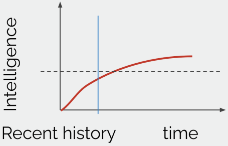
Discussion: reasoning and risk

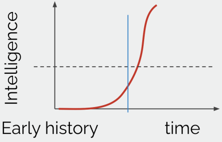
Machine reasoning hints at the idea that there is something beyond our current theory of generalisation. Do you agree?

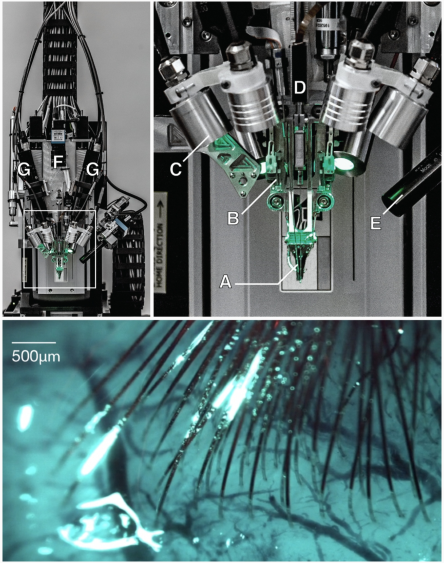


Or is reasoning just a imitation/generative modeling with representative functions?

Watch GPT-3 answer this for itself







Take Away Points

Summary

In summary, within deep learning:

- nearly all learning problems relate to generative modeling
- there's a push now towards generalising to unseen tasks
- we're heading towards a grand unification of modalities
- are reasoning and meta learning just generalisation?
- what will be the most scalable representative functions?
- humans are really bad at imagining the unknown

Visit my website for funded PhD scholarships in these areas

References I

[1] Scikitlearn. Manifold learning algorithms. Available online. 2020.

[2] Leland McInnes, John Healy, and James Melville. "Umap: Uniform manifold approximation and projection for dimension reduction". In: arXiv preprint arXiv:1802.03426 (2018).

[3] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. "Meta-learning in neural networks: A survey". In: arXiv preprint arXiv:2004.05439 (2020).

[4] Elon Musk et al. "An integrated brain-machine interface platform with thousands of channels". In: Journal of medical Internet research 21.10 (2019), e16194.