Transfer Learning

ImageNet Dataset

- ImageNet is a dataset of over **15 million labelled** high-resolution images belonging to roughly **22,000 categories**.

ImageNet Dataset

- ImageNet is a dataset of over **15 million labelled** high-resolution images belonging to roughly **22,000 categories**.

- The images were collected from the web and labelled by human labellers using Amazon's Mechanical Turk crowd-sourcing tool.

ImageNet Dataset

- ImageNet is a dataset of over **15 million labelled** high-resolution images belonging to roughly **22,000 categories**.

- The images were collected from the web and labelled by human labellers using Amazon's Mechanical Turk crowd-sourcing tool.

- Starting in 2010, as part of the Pascal Visual Object Challenge, an annual competition called the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) has been held.

ImageNet Dataset

- ImageNet is a dataset of over **15 million labelled** high-resolution images belonging to roughly **22,000 categories**.

- The images were collected from the web and labelled by human labellers using Amazon's Mechanical Turk crowd-sourcing tool.

- Starting in 2010, as part of the Pascal Visual Object Challenge, an annual competition called the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) has been held.

- **ILSVRC uses a subset of ImageNet with roughly 1000 images in each of 1000 categories.**

ImageNet Dataset

- ImageNet is a dataset of over **15 million labelled** high-resolution images belonging to roughly **22,000 categories**.

- The images were collected from the web and labelled by human labellers using Amazon's Mechanical Turk crowd-sourcing tool.

- Starting in 2010, as part of the Pascal Visual Object Challenge, an annual competition called the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) has been held.

- **ILSVRC uses a subset of ImageNet with roughly 1000 images in each of 1000 categories.**

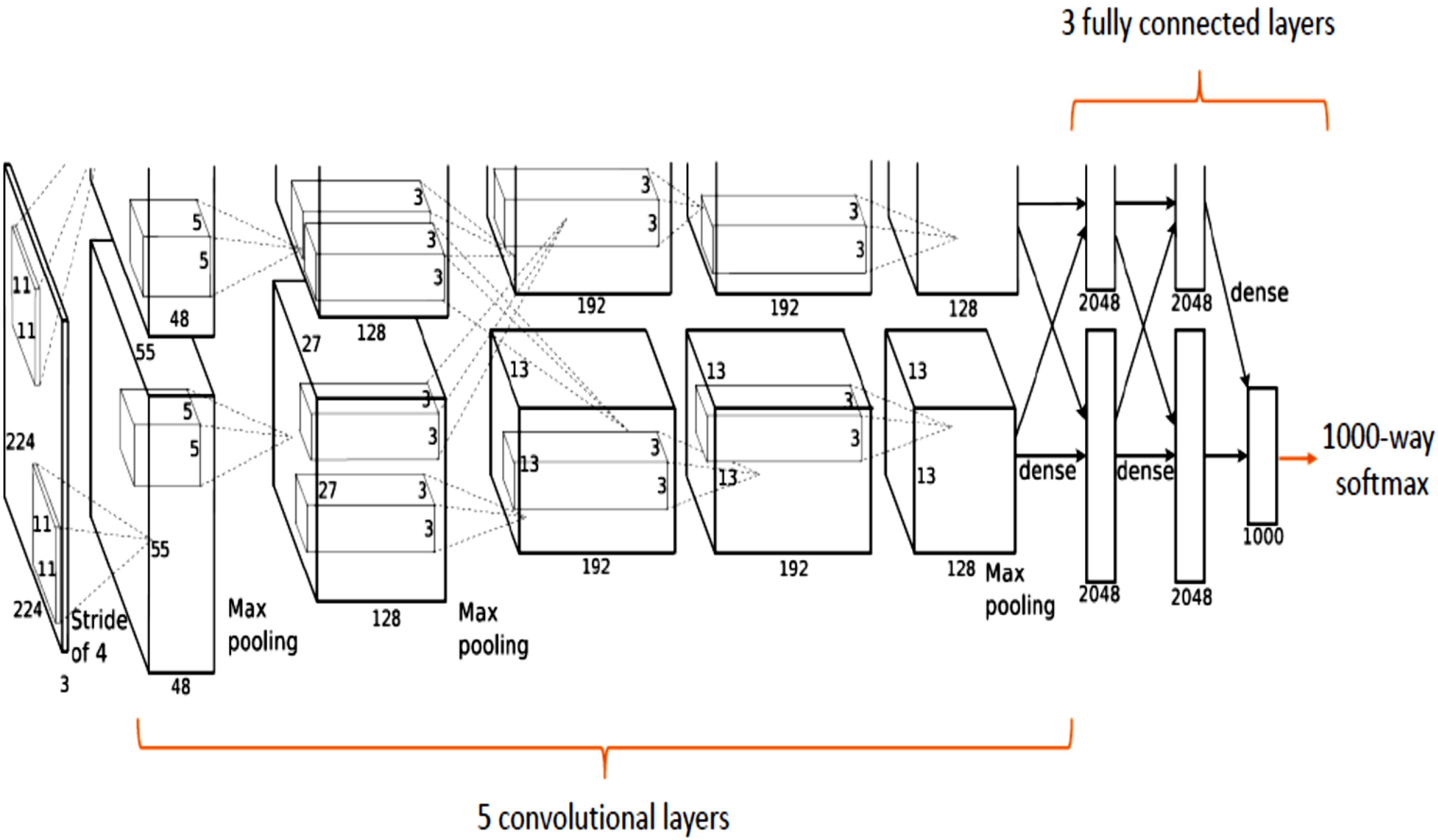
- **In all, there are roughly 1.2 million training images, 50,000 validation images, and 150,000 testing images.**

ImageNet Dataset

- **ILSVRC-2010 is the only version of ILSVRC for which the test set labels are available.**

- On ImageNet, it is customary to report two error rates: **top-1 and top-5**, where the top-5 error rate is the fraction of test images for which the correct label is not among the five labels considered most probable by the model.

ALEXNET - Architecture



These slides are not original and have been prepared from various sources for teaching purpose.

Transfer Learning

Introduction [1]

- The ability of a system **to recognize and apply knowledge and skills** learned in previous tasks to novel tasks (in new domains).

Introduction [1]

- The ability of a system **to recognize and apply knowledge and skills** learned in previous tasks to novel tasks (in new domains).

- **It is motivated by human learning. People can often transfer knowledge learnt previously to novel situations**
  - **Chess -> Checkers**
  - **Mathematics -> Computer Science**
  - **Table Tennis -> Tennis**

The traditional supervised learning setup in ML [2]

- The **traditional supervised learning paradigm breaks down when we do not have sufficient labelled data** for the task or domain we care about to train a reliable model.

The traditional supervised learning setup in ML [2]

- The **traditional supervised learning paradigm breaks down when we do not have sufficient labeled data** for the task or domain we care about to train a reliable model.

- **If we want to train a model to detect pedestrians on night-time images, we could apply a model that has been trained on a similar domain, e.g. on day-time images**.

The traditional supervised learning setup in ML [2]

- The **traditional supervised learning paradigm breaks down when we do not have sufficient labeled data** for the task or domain we care about to train a reliable model.

- **If we want to train a model to detect pedestrians on night-time images, we could apply a model that has been trained on a similar domain, e.g. on day-time images**.

- In practice, however, we often experience a **deterioration or collapse in performance** as the model has inherited the bias of its training data and does not know how to generalize to the new domain.

The traditional supervised learning setup in ML [2]

- The **traditional supervised learning paradigm breaks down when we do not have sufficient labeled data** for the task or domain we care about to train a reliable model.

- **If we want to train a model to detect pedestrians on night-time images, we could apply a model that has been trained on a similar domain, e.g. on day-time images**.

- In practice, however, we often experience a **deterioration or collapse in performance** as the model has inherited the bias of its training data and does not know how to generalize to the new domain.

- If we want to train a model to perform **a new task, such as detecting bicyclists, we cannot even reuse an existing model, as the labels between the tasks differ.**

The Transfer Learning Setup [2]

- **Transfer learning allows us to deal with these scenarios by leveraging the already existing labelled data of some related task or domain.**

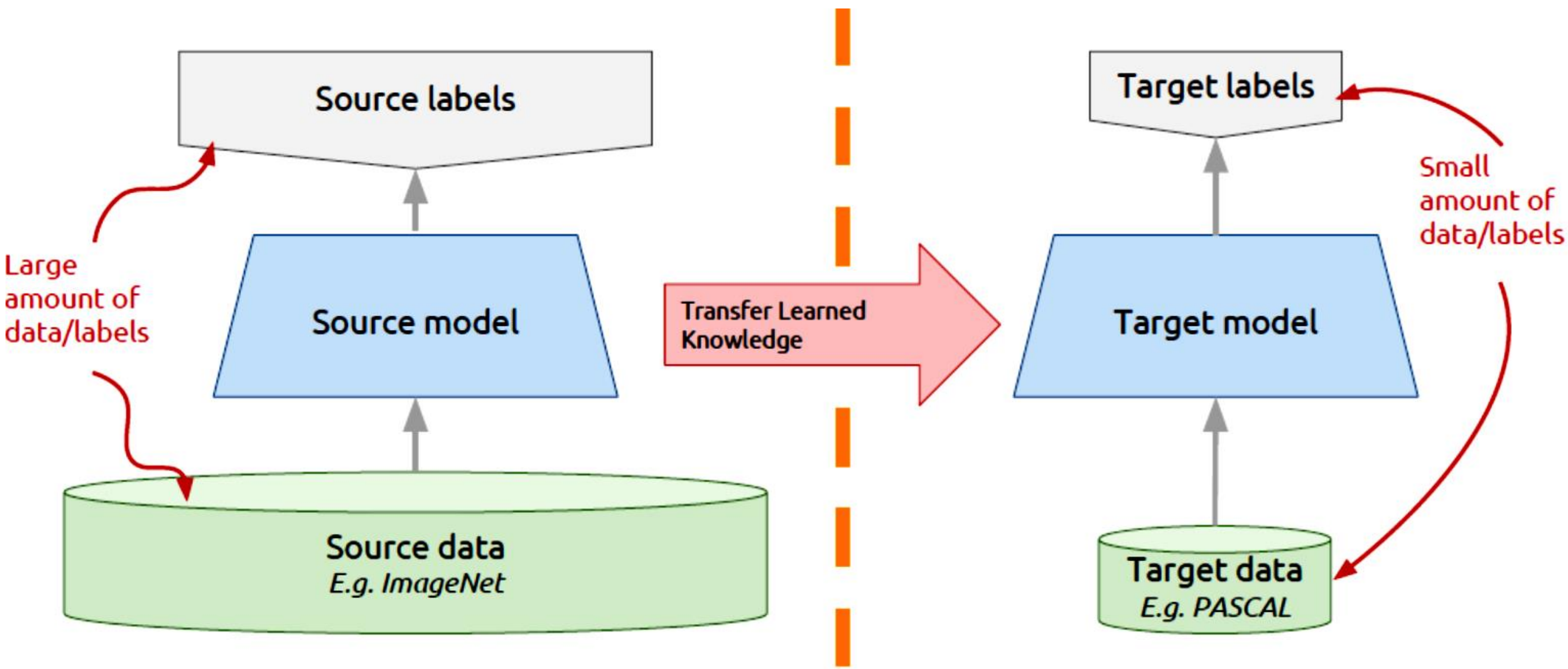
The Transfer Learning Setup [2]

- **Transfer learning allows us to deal with these scenarios by leveraging the already existing labeled data of some related task or domain.**

- **We try to store this knowledge gained in solving the source task in the source domain and apply it to our problem of interest.**

Transfer Learning: Idea [3]

- Instead of training a deep network from scratch for your task:
  - Take a network trained on a different domain for a different source task
  - Adapt it for your domain and your target task

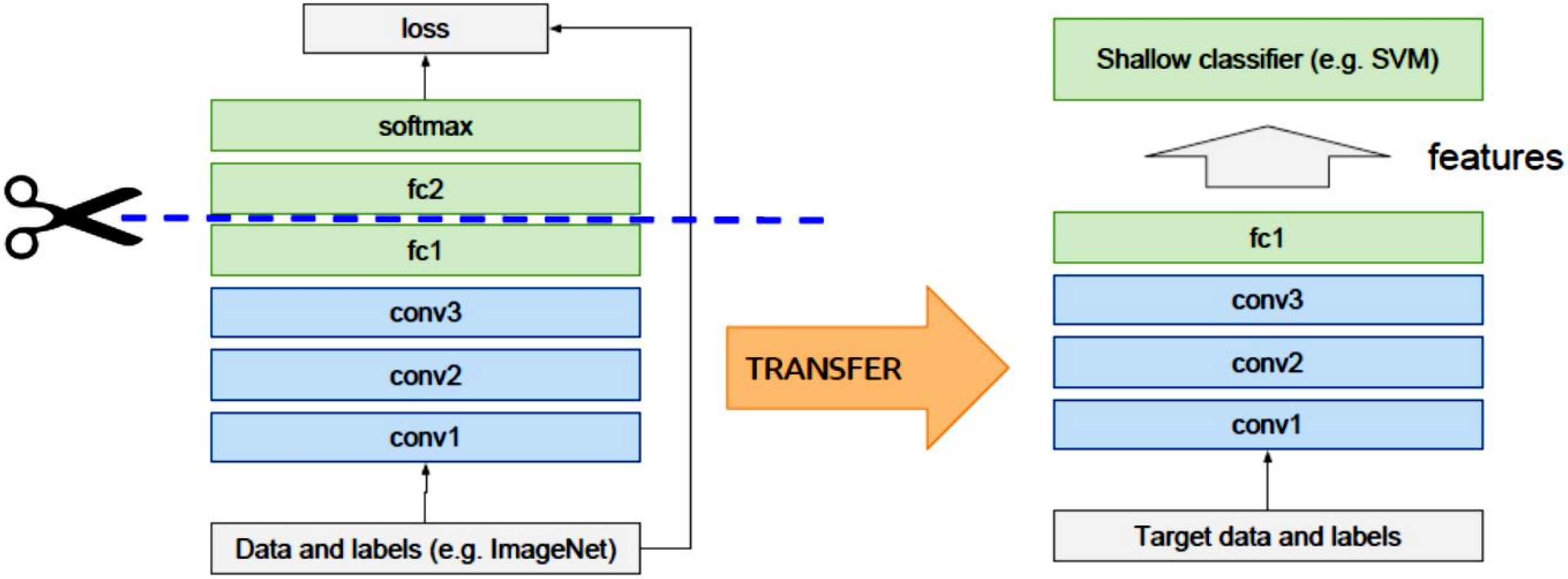


Example: PASCAL VOC 2007 [3]

- Standard classification benchmark, 20 classes, ~10K images, 50% train, 50% test
- Deep networks can have many parameters (e.g. 60M in Alexnet)
- Direct training (from scratch) using only 5K training images can be problematic. Model overfits.
- How can we use deep networks in this setting?

Off-the-shelf Features [3]

Idea: use outputs of one or more layers of a network trained on a different task as generic feature detectors. Train a new shallow model on these features.



Off-the-shelf Features [3]

Works surprisingly well in practice!

Surpassed or on par with state-of-the-art in several tasks in 2014

Image classification:

- PASCAL VOC 2007
- Oxford flowers
- CUB Bird dataset
- MIT indoors

Image retrieval:

- Paris 6k
- Holidays
- UKBench

Method	mean Accuracy
HSV [27]	43.0
SIFT internal [27]	55.1
SIFT boundary [27]	32.0
HOG [27]	49.6
HSV+SIFTi+SIFTb+HOG(MKL) [27]	72.8
BOW(4000) [14]	65.5
SPM(4000) [14]	67.4
FLH(100) [14]	72.7
BiCos seg [7]	79.4
Dense HOG+Coding+Pooling[2] w/o seg	76.7
Seg+Dense HOG+Coding+Pooling[2]	80.7
CNN-SVM w/o seg	74.7
CNNaug-SVM w/o seg	**86.8**

Oxford 102 flowers dataset

Razavian et al, **CNN Features off-the-shelf: an Astounding Baseline for Recognition,** CVPRW 2014 http://arxiv.org/abs/1403.6382

Major Transfer Learning Scenarios [5]

- In practice, very few people train an entire Convolutional Network from scratch (with random initialization), because it is relatively rare to have a dataset of sufficient size.

Major Transfer Learning Scenarios [5]

- In practice, very few people train an entire Convolutional Network from scratch (with random initialization), because it is relatively rare to have a dataset of sufficient size.

- Instead, it is common to pretrain a ConvNet on a very large dataset (e.g. ImageNet, which contains 1.2 million images with 1000 categories), and then use the ConvNet either as an initialization or a fixed feature extractor for the task of interest.

Major Transfer Learning Scenarios [5]

- The Two Major Transfer Learning scenarios look as follows:
  - **ConvNet as fixed feature extractor**
  - **Fine-tuning the ConvNet**

Major Transfer Learning Scenarios [5]

- The Two Major Transfer Learning scenarios look as follows:
  - ConvNet as fixed feature extractor
    - Take a ConvNet pretrained on ImageNet, remove the last fully-connected layer (this layer's outputs are the 1000 class scores for a different task like ImageNet), then treat the rest of the ConvNet as a fixed feature extractor for the new dataset.

Major Transfer Learning Scenarios [5]

- The Two Major Transfer Learning scenarios look as follows:
  - ConvNet as fixed feature extractor
    - Take a ConvNet pretrained on ImageNet, remove the last fully-connected layer (this layer's outputs are the 1000 class scores for a different task like ImageNet), then treat the rest of the ConvNet as a fixed feature extractor for the new dataset.
    - In an AlexNet, this would compute a 4096-D vector for every image that contains the activations of the hidden layer immediately before the classifier. We call these features CNN codes.

Major Transfer Learning Scenarios [5]

- The Two Major Transfer Learning scenarios look as follows:
  - ConvNet as fixed feature extractor
    - Take a ConvNet pretrained on ImageNet, remove the last fully-connected layer (this layer's outputs are the 1000 class scores for a different task like ImageNet), then treat the rest of the ConvNet as a fixed feature extractor for the new dataset.
    - In an AlexNet, this would compute a 4096-D vector for every image that contains the activations of the hidden layer immediately before the classifier. We call these features CNN codes.
    - It is important for performance that these codes are ReLUd (i.e. thresholded at zero) if they were also thresholded during the training of the ConvNet on ImageNet (as is usually the case).

Major Transfer Learning Scenarios [5]

- The Two Major Transfer Learning scenarios look as follows:
  - ConvNet as fixed feature extractor
    - Take a ConvNet pretrained on ImageNet, remove the last fully-connected layer (this layer's outputs are the 1000 class scores for a different task like ImageNet), then treat the rest of the ConvNet as a fixed feature extractor for the new dataset.
    - In an AlexNet, this would compute a 4096-D vector for every image that contains the activations of the hidden layer immediately before the classifier. We call these features CNN codes.
    - It is important for performance that these codes are ReLUd (i.e. thresholded at zero) if they were also thresholded during the training of the ConvNet on ImageNet (as is usually the case).
    - Once you extract the 4096-D codes for all images, train a linear classifier (e.g. Linear SVM or Softmax classifier) for the new dataset.

Major Transfer Learning Scenarios [5]

- The Two Major Transfer Learning scenarios look as follows:
  - Fine-tuning the ConvNet
    - The second strategy is to not only replace and retrain the classifier on top of the ConvNet on the new dataset, but to also fine-tune the weights of the pretrained network by continuing the backpropagation.

Major Transfer Learning Scenarios [5]

- The Two Major Transfer Learning scenarios look as follows:
  - Fine-tuning the ConvNet
    - The second strategy is to not only replace and retrain the classifier on top of the ConvNet on the new dataset, but to also fine-tune the weights of the pretrained network by continuing the backpropagation.
    - **It is possible to fine-tune all the layers of the ConvNet, or it's possible to keep some of the earlier layers fixed (due to overfitting concerns) and only fine-tune some higher-level portion of the network.**

Major Transfer Learning Scenarios [5]

- The Two Major Transfer Learning scenarios look as follows:
  - Fine-tuning the ConvNet
    - The second strategy is to not only replace and retrain the classifier on top of the ConvNet on the new dataset, but to also fine-tune the weights of the pretrained network by continuing the backpropagation.

    - **It is possible to fine-tune all the layers of the ConvNet, or it's possible to keep some of the earlier layers fixed (due to overfitting concerns) and only fine-tune some higher-level portion of the network.**

    - **This is motivated by the observation that the earlier features of a ConvNet contain more generic features (e.g. edge detectors or colour blob detectors) that should be useful to many tasks, but later layers of the ConvNet becomes progressively more specific to the details of the classes contained in the original dataset.**

When and How to Fine Tune? [5]

- **How do you decide what type of transfer learning you should perform on a new dataset?**

When and How to Fine Tune? [5]

- **How do you decide what type of transfer learning you should perform on a new dataset?**

- **This is a function of several factors, but the two most important ones are the size of the new dataset (small or big), and its similarity to the original dataset** (e.g. ImageNet-like in terms of the content of images and the classes, or very different, such as microscope images).

When and How to Fine Tune? [5]

- **How do you decide what type of transfer learning you should perform on a new dataset?**

- **This is a function of several factors, but the two most important ones are the size of the new dataset (small or big), and its similarity to the original dataset** (e.g. ImageNet-like in terms of the content of images and the classes, or very different, such as microscope images).

- **Keeping in mind that ConvNet features are more generic in early layers and more original-dataset-specific in later layers, here are some common rules of thumb for navigating the 4 major scenarios:**
  - **New dataset is small and similar to original dataset**
  - **New dataset is large and similar to the original dataset**
  - **New dataset is small but very different from the original dataset**
  - **New dataset is large and very different from the original dataset**

When and How to Fine Tune? [5]

- New dataset is small and similar to original dataset

When and How to Fine Tune? [5]

- New dataset is small and similar to original dataset
  - Since the data is small, it is not a good idea to fine-tune the ConvNet due to overfitting concerns.

When and How to Fine Tune? [5]

- New dataset is small and similar to original dataset
  - Since the data is small, it is not a good idea to fine-tune the ConvNet due to overfitting concerns.
  - Since the data is similar to the original data, we expect higher-level features in the ConvNet to be relevant to this dataset as well.

When and How to Fine Tune? [5]

- New dataset is small and similar to original dataset
  - Since the data is small, it is not a good idea to fine-tune the ConvNet due to overfitting concerns.
  - Since the data is similar to the original data, we expect higher-level features in the ConvNet to be relevant to this dataset as well.
  - Hence, the best idea might be to train a linear classifier on the CNN codes.

When and How to Fine Tune? [5]

- New dataset is large and similar to the original dataset

When and How to Fine Tune? [5]

- New dataset is large and similar to the original dataset
  - Since we have more data, we can have more confidence that we won't overfit if we were to try to fine-tune through the full network.

When and How to Fine Tune? [5]

- New dataset is small but very different from the original dataset

When and How to Fine Tune? [5]

- New dataset is small but very different from the original dataset
  - Since the data is small, it is likely best to only train a linear classifier.

When and How to Fine Tune? [5]

- New dataset is small but very different from the original dataset
  - Since the data is small, it is likely best to only train a linear classifier.
  - Since the dataset is very different, it might not be best to train the classifier form the top of the network, which contains more dataset-specific features.

When and How to Fine Tune? [5]

- New dataset is small but very different from the original dataset
  - Since the data is small, it is likely best to only train a linear classifier.
  - Since the dataset is very different, it might not be best to train the classifier form the top of the network, which contains more dataset-specific features.
  - Instead, it might work better to train the SVM classifier from activations somewhere earlier in the network.

When and How to Fine Tune? [5]

- New dataset is large and very different from the original dataset

When and How to Fine Tune? [5]

- New dataset is large and very different from the original dataset
  - Since the dataset is very large, we may expect that we can afford to train a ConvNet from scratch.

When and How to Fine Tune? [5]

- New dataset is large and very different from the original dataset
  - Since the dataset is very large, we may expect that we can afford to train a ConvNet from scratch.
  - However, in practice it is very often still beneficial to initialize with weights from a pretrained model.

When and How to Fine Tune? [5]

- New dataset is large and very different from the original dataset
  - Since the dataset is very large, we may expect that we can afford to train a ConvNet from scratch.
  - However, in practice it is very often still beneficial to initialize with weights from a pretrained model.
  - In this case, we would have enough data and confidence to fine-tune through the entire network.

Negative Transfer [11]

- **Negative transfer happens when source domain data and task contribute to reduced performance of learning in the target domain.**

- **Causes:**
  - **Domains are too dissimilar.**
  - **Tasks are not well-related**

References

1. www.ntu.edu.sg/home/sinnopan/.../A%20Survey%20on%20Transfer%20Learning.ppt

2. http://ruder.io/transfer-learning/

3. http://imatge-upc.github.io/telecombcn-2016-dlcv/slides/D2L5-transfer.pdf

4. Olivas, Emilio Soria, ed. Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques: Algorithms, Methods, and Techniques. IGI Global, 2009.

5. http://cs231n.github.io/transfer-learning/

References

6. https://machinelearningmastery.com/transfer-learning-for-deep-learning/

7. Rusu, A. A., Vecerik, M., Rothörl, T., Heess, N., Pascanu, R., & Hadsell, R. (2016). Sim-to-Real Robot Learning from Pixels with Progressive Nets. arXiv Preprint arXiv:1610.04286.

8. Johnson, M., Schuster, M., Le, Q. V, Krikun, M., Wu, Y., Chen, Z., … Dean, J. (2016). Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation.

9. Sun, B., Feng, J., & Saenko, K. (2016). Return of Frustratingly Easy Domain Adaptation. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)

References

10. https://en.wikipedia.org/wiki/Domain_adaptation

11. https://pdfs.semanticscholar.org/presentation/f4af/de757b9dfc697d149e95cb193aa4749530e2.pdf

12. Ganin, Yaroslav, and Victor Lempitsky. "Unsupervised domain adaptation by backpropagation." arXiv preprint arXiv:1409.7495 (2014).

13. https://medium.com/@2017csm1006/unsupervised-domain-adaptation-by-backpropagation-da730a190fd2

Disclaimer

- These slides are not original and have been prepared from various sources for teaching purpose.