Explicit Inductive Bias for Transfer Learning with Convolutional Networks

Hao LIU, Linxiao ZENG, Zhufeng LI

Paris-Sud University

February 7, 2019

Overview

1. Introduction
2. Regularizers for Fine-Tuning
3. Experiments
4. Conclusion

Transfer Learning

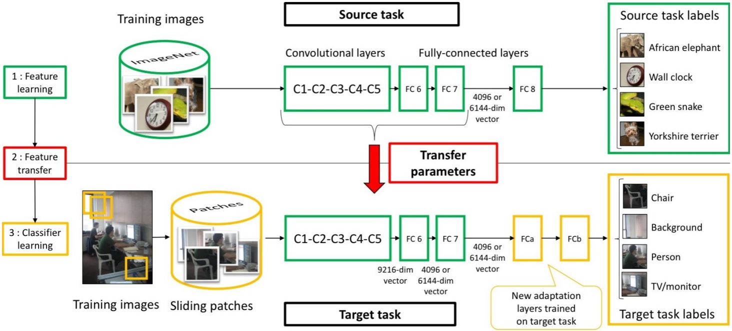

Figure: A typical transfer learning scenario

Why using TL

Deep learning models are layered architectures that learn different features at different layers.The initial layers have been seen to capture generic features, while the later ones focus more on the specific task at hand.

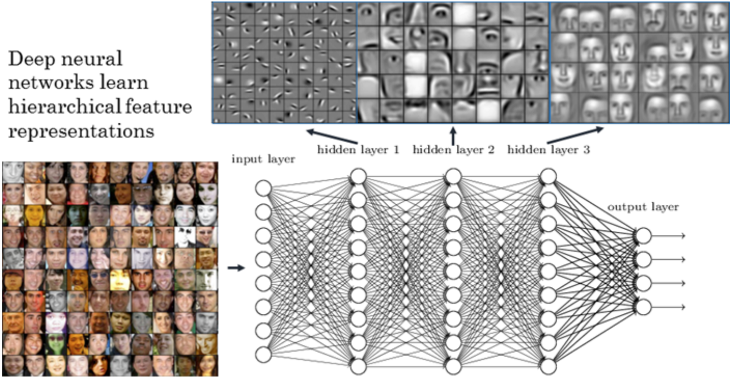

Figure: Deep Neural Network as Feature Extractor

Fine-tune or Freeze

Which to take depends on target task:

1. Fine-tune: target task classes are more abundant, need to learn more new feature
2. Freeze: target task classes are scarce, need to avoid overfitting

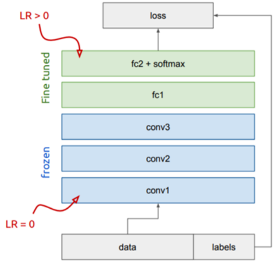

Figure: Fine-tune vs Freeze

In this paper, we will focus on fine-tuning.

Problem in Fine-tuning

A default fine-tuning setting:

1. Initialize with pretrained parameter
2. adapt to target task by minimize loss and L^2 regularization with early stop

Problem:

- parameters may be **driven far away** from their initial values!
- **losses of the initial knowledge!**

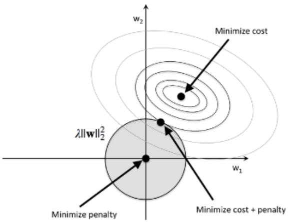

Figure: L^2 regularization

Question:

- **Is it consistent to use L^2 in transfer learning scenario?**

Why regularizers?

- Facilitating optimization and avoiding overfitting
- Pertaining to the source problem (domain and task)

Regularized objective function

$$\widetilde{J} = J + \Omega(w)$$

- Explicit inductive bias towards the initial solution
- $\Omega(w)$ act as log prior, $\widetilde{J}$ MAP estimation

L^2 regularization

$$\Omega(\omega) = \frac{\alpha}{2} \|w\|_2^2$$

L^2-SP regularization

$$\Omega(w) = \frac{\alpha}{2} \left\| w_s - w_s^0 \right\|_2^2 + \frac{\beta}{2} \|w_{\bar{s}}\|_2^2$$

L^2-SP-Fisher regularization

$$\Omega(w) = \frac{\alpha}{2} \sum_{j \in S} \hat{F}_{jj} (w_s - w_s^0)^2 + \frac{\beta}{2} \|w_{\bar{s}}\|_2^2$$

L^1-SP regularization

$$\Omega(w) = \alpha \left\| w_s - w_s^0 \right\|_1 + \frac{\beta}{2} \left\| w_{\bar{s}} \right\|_2^2$$

Group-Lasso-SP regularization

$$\Omega(w) = \alpha \sum_{g=1}^{G} s_g \left\| w_{g_g} - w_{g_g}^0 \right\|_2^2 + \frac{\beta}{2} \left\| w_{\bar{s}} \right\|_2^2$$

Group-Lasso-SP-Fisher

$$\Omega(w) = \alpha \sum_{g=1}^{G} s_g \Big(\sum_{j \in \mathcal{G}_g} \hat{F}_{jj} (w_s - w_s^0)^2 \Big)^{\frac{1}{2}} + \frac{\beta}{2} \left\| w_{\mathcal{G}_0} \right\|_2^2$$

Databases

Database	task category	traning	test	classes
MIT Indoors 67	scene classification	80	20	67
Standford Dogs 120	specific object recog.	100	72	120
Caltech 256-30	generic object recog.	30	20	257
Caltech 256-60	generic object recog.	60	20	257

Table: Characteristics of the target databases

ResNet101 as base network

Accuracy

	MIT Indoors 67	Stanford Dogs 120	Caltech 256 30	Caltech 256 60
L^2	79.6± 0.5	81.± 0.2	81.5± 0.2	85.3± 0.2
L^2-SP	84.2± 0.3	85.1± 0.2	83.5± 0.1	86.4± 0.2
L^2-SP-Fisher	84.0± 0.4	85.1± 0.2	83.3± 0.1	86.0± 0.1

Table: Average classification accuracies (in %) of L^2, L^2-SP and L^2-SP-Fisher

Results

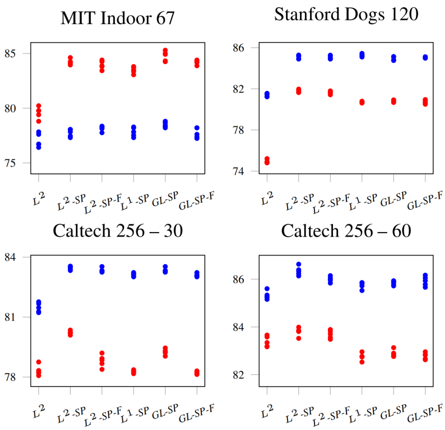

Figure: Classification accuracies (in %) of the tested fine-tuning approaches on the four target databases, using ImageNet (dark blue dots) or Places 365 (light red dots) as source databases. MIT Indoor 67 is similar to Places 365; Stanford Dogs 120 and Caltech 256 are similar to ImageNet.

Accuracy drop

Database	L^2	L^2-**SP**	L^2-**SP-Fisher**
MIT Indoors 67	-24.1	-5.3	-4.9
Standford Dogs 120	-14.1	-4.7	-4.2
Caltech 256-30	-15.4	-4.2	-3.6
Caltech 256-60	-16.9	-3.6	-3.2

Table: Classification accuracy drops (in %) on the source tasks due to fine-tuning based on L^2, L^2-SP and L^2-SP-Fisher regularizers.

Problem

Clarity of paper

1. Learning rates
2. Performance drop

Our implementation

1. Too much data
2. Expensive training
3. Very limited GPU access

Conlusion

- L^2-SP VS L^2-SP-Fisher Occam's Razor
- L^2-SP retains the features learned on source databases
- L^2-SP as a standard baseline in inductive transfer learning