Motivation
When expanding the capability of a vision system...
- Fine-tuning? (old task suffers)
- Joint training:
  - Training data required again?

What if the original dataset...
- Is not recorded?
- Is proprietary?
- Is too cumbersome?

But we want...
- Benefit of shared representation
- No or little degradation of the original capability
- Without the need to access original task dataset?

Goal:
Add new capabilities to a CNN-based vision system using only data from the new task.

Our strengths:
- Outperforms the widely-used fine-tuning on both original and new task.
- Outperforms feature extraction on the new task.
- Simple to implement and deploy
- Training efficiency compared to joint training

Method
Outline
1. Obtain old task responses
   - [old task 1 response $Y_o^1$]
   - [old task m response $Y_o^m$]
   - shared parameters
   - task-specific
2. Train on new images
   - [new image]
   - [new task 1 response $Y_n^1$]
   - [new task m response $Y_n^m$]
   - new task ground truth $Y_n$

Training: loss
shared/old/new parameters $\theta^o, \theta^e, \theta^s \leftarrow \argmin_{\theta^o, \theta^e, \theta^s} \sum_i L_{old}(Y_o^i, Y_e^i, Y_n^i) + L_{new}(Y_n^i, Y_e^i, Y_n^i) + \lambda (\theta_o, \theta_e, \theta_s)$

Limitations of existing methods

Experiment Settings
Datasets
- old task
  - ImageNet2012*
  - Places2*
- new task
  - PASCAL VOC 2012
  - Caltech-UCSD Birds
  - MIT indoor scenes
  - MNIST

* Pre-trained AlexNet obtained from authors

Efficiency:
- Training: forward-pass shared parameters once. Faster than joint training, similar to fine-tuning
- Test: same as compared methods: more efficient than keeping different networks for each task.

Design choices and alternatives
We experimented with some variations:
- Possibly: more layers as task-specific parameters.
- Possibly: add nodes to earlier layers
- Possibly: use alternative loss for $L_{old}(Y_o, Y_e)$
- Possibly: just reduce fine-tuning learning rate

These variations provided insignificant or inconsistent improvements, if any.

Conclusions
- Vs. Feature Extraction: LwF outperforms on new task; underperforms on old task
- Vs. Fine-tuning: LwF outperforms on both tasks, as keeping old responses regularizes model
- Vs. Joint Training: LwF performs nearly as well as joint training
- Dissimilar new tasks degrade old task performance
- Similar results and some observations for adding multiple new tasks

Acknowledgments
This material is based upon work supported in part by NSF Awards 14-46765 and 10-53768 and ONR MURI N00014-16-1-2007.