(Towards) Lifelong Machine Learning

Christoph Lampert

usually:

currently:
IST Austria (Institute of Science and Technology Austria)

- public research institute
- opened in 2009
- located in outskirts of Vienna

Focus on Basic Research
- curiosity-driven
- foster interdisciplinarity
- currently 45 research groups:
  - Computer Science, Mathematics, Physics, Biology, Neuroscience

We’re hiring on all levels:
- from interns ... to full professors
Overview of the research in our group

Theory (Statistical Machine Learning)
- Multi-task learning
- Domain adaptation
- Lifelong learning / learning to learn
- Learning with dependent data

Models/Algorithms
- Zero-shot learning
- Incremental learning
- Weakly-supervised learning
- Structured prediction / graphical models

Applications (in Computer Vision)
- Object recognition
- Image generation
- Image segmentation
- Semantic image representations
Overview of the research in our group

Theory (Statistical Machine Learning)
- Multi-task learning
- Domain adaptation
- Lifelong learning / learning to learn
- Learning with dependent data

Models/Algorithms
- Zero-shot learning
- Incremental learning
- Weakly-supervised learning
- Structured prediction / graphical models

Applications (in Computer Vision)
- Object recognition
- Image generation
- Image segmentation
- Semantic image representations
Each task is learned in isolation
Lifelong learning

Later tasks can exploit information extracted from earlier tasks e.g. [Thrun 1996]
What can we do well?

- **multi-task learning** [Caruana, ML 1997]
  - given several tasks, all of which are relevant
  - solve all tasks jointly

- **task-to-task transfer** e.g. [Daume III & Marcu, JAIR 2006]
  - given two tasks:
    - one that is meant to be solved (*target*), and
    - one that can be used to do so (*source*)
  - solve the *target* using information from the *source*
What can’t we do well?

- benefit from task chains longer than 2
- automatically determine which task to solve next
- extract knowledge without knowing what it should be used for in the future

... and a lot more.
iCaRL: Incremental Classifier and Representation Learning

CVPR 2017
arXiv:1611.07725 [cs.CV]
Class-incremental Learning

Setup: training data arrives one class at a time
the system can only store a (small) fixed-size amount of it
Goal: learn multi-class classifier, but avoid catastrophic forgetting

Images: CIFAR dataset
Class-incremental Learning

Method:

● train **fixed-size deep network** with ordinary BackProp
  ○ all weights are trained in all steps, no *freezing* of layers
  ○ network size remains constant, no layers/columns added

● keep a small **set of exemplars** from all classes
  ○ greedy procedure to select exemplars, inspired by *herding* [Welling, ICML 2009]
  ○ ability to remove exemplars on-the-fly

● ‘**nearest-mean-of-exemplars’ classifier**, inspired by NCM [Mensink et al, TPAMI 2013]
  ○ adjusts automatically whenever underlying data representation changes
  ○ more robust than network outputs

● **add distillation term** to loss function, similar to *Learning without Forgetting* [Li&Hoiem, ECCV 2016]
  ○ stabilizes outputs from one epoch to the next
  ○ limited overhead, no need to store multiple copies of the network
Class-incremental Learning

Results:
- iCaRL learns reasonably well
- other methods fail quickly

Observations:
- class-incremental learning is hard
- catastrophic forgetting is real
- distillation as regularizer is not enough
- exemplars help a lot
Class-incremental Learning

Results:
- iCaRL learns reasonably well
- other methods fail quickly

Observations:
- class-incremental learning is hard
- catastrophic forgetting is real
- distillation as regularizer is not enough
- exemplars help a lot

batch training (everything shuffled i.i.d.): 68.2%

Multi-class accuracies over 10 repeats (average and standard deviation) for class-incremental training on CIFAR-100

Confusion matrices of different method on CIFAR-100 after training for 100 classes with 10 classes per batch
Classifier Adaptation at Prediction Time

CVPR 2015

Amelie Royer (IST Austria)
Classifier Adaptation

Training time:
- classes are balanced
- data is i.i.d.

Prediction time:
- different class proportions,
- time-varying data distribution,
- data has strong dependencies
Classifier Adaptation

Method:
- estimate class proportions of test data on-the-fly
- adapt classifier outputs to reflect current class priors \( \pi \)

**Definition:** The class-prior adaptation of \( f \) from \( \rho \) to \( \pi \) is

\[
g(x) = \arg\max_{y \in Y} g_y(x) \quad \text{for} \quad g_y(x) = \frac{f_y(x)\pi_y}{\rho_y}.
\]

[Saerens et al., 2002]

Different kinds of feedback signal:
- online: after each prediction the true class label is revealed
- bandit: after each prediction, if it revealed if the prediction was correct or not
- unsupervised: no information available
Classifier Adaptation

**Results:**

- adaptation helps
- if base classifiers are reasonable, improved accuracy, even without feedback (unsupervised)
- only small degradation, if there is no signal to adapt to (RND)

<table>
<thead>
<tr>
<th></th>
<th>ILSVRC2012</th>
<th>CNN</th>
<th>CNN+adapt</th>
<th>CNN+dyn</th>
</tr>
</thead>
<tbody>
<tr>
<td>TXT</td>
<td>19.8 ± 1.9</td>
<td>14.8 ± 1.8</td>
<td>16.4 ± 1.7</td>
<td></td>
</tr>
<tr>
<td>MDS</td>
<td>16.1 ± 6.5</td>
<td>6.8 ± 3.6</td>
<td>8.1 ± 2.9</td>
<td></td>
</tr>
<tr>
<td>MDS(0.001)</td>
<td>15.6 ± 4.5</td>
<td>9.0 ± 3.4</td>
<td>9.4 ± 2.7</td>
<td></td>
</tr>
<tr>
<td>MDS(0.01)</td>
<td>15.7 ± 1.8</td>
<td>13.6 ± 1.7</td>
<td>11.3 ± 1.5</td>
<td></td>
</tr>
<tr>
<td>MDS(0.1)</td>
<td>16.2 ± 0.8</td>
<td>16.8 ± 0.8</td>
<td>14.9 ± 0.8</td>
<td></td>
</tr>
<tr>
<td>KS</td>
<td>16.4 ± 1.8</td>
<td>16.5 ± 1.7</td>
<td>14.2 ± 1.5</td>
<td></td>
</tr>
<tr>
<td>KS(0.001)</td>
<td>16.5 ± 1.8</td>
<td>16.6 ± 1.9</td>
<td>14.1 ± 1.5</td>
<td></td>
</tr>
<tr>
<td>KS(0.01)</td>
<td>16.4 ± 1.4</td>
<td>16.7 ± 1.5</td>
<td>14.1 ± 1.1</td>
<td></td>
</tr>
<tr>
<td>KS(0.1)</td>
<td>16.5 ± 0.8</td>
<td>17.4 ± 0.8</td>
<td>15.2 ± 0.8</td>
<td></td>
</tr>
<tr>
<td>RND</td>
<td>16.5 ± 0.6</td>
<td>18.0 ± 0.7</td>
<td>16.9 ± 0.6</td>
<td></td>
</tr>
</tbody>
</table>

Unsupervised (No Feedback)

Values: top-5 classification accuracy over 100 runs on different data sequences
Multitask Learning with Labeled and Unlabeled Tasks

ICML 2017
arXiv:1602.06518 [stat.ML]
Active Task Selection for Multitask Learning with Labeled and Unlabeled Tasks

Setup: many supervised learning tasks, but all have only unlabeled data
the system may ask for labels for a (small) subset of tasks

Goal: learn predictors for all tasks (labeled and unlabeled ones)
Active Task Selection for Multitask Learning with Labeled and Unlabeled Tasks

Method: from theory to algorithm

- **generalization bound**, inspired by multi-source domain adaptation [Ben-David et al, ML 2010]

\[
\begin{align*}
\sum_{t=1}^{T} \sum_{i=1}^{k} \alpha_i \hat{e}_i(h_t) & \leq \sum_{t=1}^{T} \sum_{i=1}^{k} \alpha_i \hat{e}_i(h_t) + \sum_{t=1}^{T} \sum_{i=1}^{k} \alpha_i \text{disc}(S_t, S_i) + \sum_{t=1}^{T} \sum_{i=1}^{k} \alpha_i \lambda_{ti} + A\|\alpha\|_{1,2} + B\|\alpha\|_{1,2} + \ldots
\end{align*}
\]

- **minimizing the bound** yields
  - w.r.t. \( \alpha_i \) amount of sharing between tasks and indicators which tasks to ask for labels for,
  - w.r.t. \( h_t \) predictors for all tasks (learned from suitably weighted labeled data)
Active Task Selection for Multitask Learning with Labeled and Unlabeled Tasks

Method: from theory to algorithm

- generalization bound, inspired by multi-source domain adaptation \cite{Ben-David et al, ML 2010}
  - minimizing the bound yields
    - \[ \sum_{t=1}^{T} \alpha_t \sum_{i=1}^{k} \alpha_i \text{er}_i(h_t) + \sum_{t=1}^{T} \sum_{i=1}^{k} \alpha_t \lambda_i \text{er}_i(S_i) + \sum_{t=1}^{T} \sum_{i=1}^{k} \alpha_t \lambda_i \text{er}_i(S_i) + B \alpha_1 + B \alpha_2 + \ldots \]
    - w.r.t. \( \alpha_i \) amount of sharing between tasks and indicators which tasks to ask for labels for,
    - w.r.t. \( h_t \) predictors for all tasks (learned from suitably weighted labeled data)
Active Task Selection for Multitask Learning with Labeled and Unlabeled Tasks

all tasks unlabeled
determine weights: how much should any task rely on any other?
Active Task Selection for Multitask Learning with Labeled and Unlabeled Tasks

determine tasks to have labeled
Active Task Selection for Multitask Learning with Labeled and Unlabeled Tasks
Active Task Selection for Multitask Learning with Labeled and Unlabeled Tasks

learn predictors for each task (labeled and unlabeled) from weighted samples
Active Task Selection for Multitask Learning with Labeled and Unlabeled Tasks

Dataset: Multitask dataset of product reviews (MDPR)
http://cvml.ist.ac.at/MDPR/

Results:
- better to have multiple predictors than just a single one for all unlabeled tasks
- actively select tasks to be labeled improves over choosing randomly
Curriculum Learning of Multiple Tasks

CVPR 2015
arXiv:1611.07725 [stat.ML]

Asya Pentina
(IST Austria)

Viktoriia Sharmanska
(U Sussex)
Learning a Task Order

Setup: solve multiple tasks sequentially, always transfer from one to the next
Question: what is the best order to do so?
Learning a Task Order

**Method:** from theory to algorithm

- **generalization bound,** inspired by SVMs with biased regularization [Kienzle et al., ICML 2006]

\[
\frac{1}{2n} \sum_{i=1}^{n} \mathbb{E}_{(x,y) \sim D_i} \left[ y \neq \text{sign}(w_i, x) \right] \leq \frac{1}{n} \sum_{i=1}^{n} \left[ \frac{1}{m_{\pi(i)}} \sum_{j=1}^{m_{\pi(i)}} \Phi \left( \frac{y_{j}^{\pi(i)} \langle w_{\pi(i)}, x_{j}^{\pi(i)} \rangle}{\|x_{j}^{\pi(i)}\|} \right) + \frac{\|w_{\pi(i)} - w_{\pi(i-1)}\|^2}{2\sqrt{m}} \right] + \ldots
\]

- **minimizing the bound yields**
  - w.r.t. \( \pi \) order in which to learn the tasks
  - w.r.t. \( w_i \) predictors (classifiers) for each tasks

- greedy strategy: start with easy task, then move to harder but related ones
Learning a Task Order

Method: from theory to algorithm

- generalization bound, inspired by SVMs with biased regularization [Kienzle et al, ICML 2006]
  \[ \frac{1}{2n} \sum_{i=1}^{n} (x_i, y_{i}) \neq \text{sign}(\langle \omega, x \rangle + b) \]

- minimizing the bound w.r.t.
  - \( \pi \): order in which to learn the tasks
  - \( w_i \): predictors (classifiers) for each task

- greedy strategy: start with easy task, then move to harder but related ones
Results:

- multi-task better than independent learning
- sequential on-par or better to traditional

Results:

- order of tasks matters
- minimizing bound yields better order than random or heuristics

Dataset: Animals with Attributes, http://cvml.ist.ac.at/AwA/
Summary

- lifelong learning seems a “natural” way to overcome the need for huge training sets
- current machine learning systems are far from having lifelong learning capabilities
- deep learning triggered new problems
  - how to measure task similarity?
  - how to transfer knowledge beyond copying parameters?
  - how to prevent catastrophic forgetting?
- to make progress we need more:
  - theoretical understanding
  - models that can preserve knowledge
  - real-world problems / datasets
  - evaluation measures

(so essentially everything)
Summary

- lifelong learning seems a “natural” way to overcome the need for huge training sets
- current machine learning systems are far from having lifelong learning capabilities
- deep learning triggered new problems
  - how to measure task similarity?
  - how to transfer knowledge beyond copying parameters?
  - how to prevent catastrophic forgetting?
- to make progress we need more:
  - theoretical understanding
  - models that can preserve knowledge
  - real-world problems / datasets
  - evaluation measures
  (so essentially everything)
Summary

- lifelong learning seems a “natural” way to overcome the need for huge training sets
- current machine learning systems are far from having lifelong learning capabilities

- deep learning triggered new problems
  - how to measure task similarity?
  - how to transfer knowledge beyond copying parameters?
  - how to prevent catastrophic forgetting?

- to make progress we need more:
  - theoretical understanding
  - models that can preserve knowledge
  - real-world problems / datasets
  - evaluation measures

(so essentially everything)

Thanks to...

Google

Funding:

Group members and alumni:

- Sylvestre Rebuffi
- Asya Pentina
- Viktoriia Sharmanska
- Alex Kolesnikov
- Amelie Royer
- Georg Sperl
- Alex Zimin
- Mary Phuong
current machine learning systems are far from having lifelong learning capabilities

lifelong learning seems a “natural” way to overcome the need for huge training sets

deep learning triggered new problems
  ○ how to measure task similarity?
  ○ how to transfer knowledge beyond copying parameters?
  ○ how to prevent catastrophic forgetting?

to make progress we need more:
  ○ theoretical understanding
  ○ models that can preserve knowledge
  ○ real-world problems / datasets
  ○ evaluation measures

(so essentially everything)

Summary

Thanks to…

Funding:

Group members and alumni:

Sylvestre Rebuffi
Asya Pentina
Viktoriiia Sharmanska
Alex Kolesnikov
Amelie Royer
Georg Sperl
Alex Zimin
Mary Phuong
Summary

- lifelong learning seems a “natural” way to overcome the need for huge training sets
- current machine learning systems are far from having lifelong learning capabilities
- deep learning triggered new problems
  - how to measure task similarity?
  - how to transfer knowledge beyond copying parameters?
  - how to prevent catastrophic forgetting?
- to make progress we need more:
  - theoretical understanding
  - models that can preserve knowledge
  - real-world problems / datasets
  - evaluation measures
  (so essentially everything)
Summary

- lifelong learning seems a “natural” way to overcome the need for huge training sets
- current machine learning systems are far from having lifelong learning capabilities
- deep learning triggered new problems
  - how to measure task similarity?
  - how to transfer knowledge beyond copying parameters?
  - how to prevent catastrophic forgetting?
- to make progress we need more:
  - theoretical understanding
  - models that can preserve knowledge
  - real-world problems / datasets
  - evaluation measures

(so essentially everything)