Trustworthy Machine Learning

Christoph Lampert

IST Austria
Institute of Science and Technology

Renningen, May 22, 2018
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 49 research groups:
  - Computer Science, Mathematics, Physics, Biology, Neuroscience, Chemistry

Co-located Technology Park:
- up to 28,000 m$^2$ for spin-offs, startups, and company research labs

Questions? chl@ist.ac.at
"Machine learning and AI is a horizontal enabling layer. It will empower and improve every business, every government organization, every philanthropy – basically there’s no institution in the world that cannot be improved with machine learning."

Jeff Bezos, 2017
Artificial Intelligence is in a Crisis
Artificial Intelligence is in a Crisis

“I would trust a fully autonomous vehicle.”
Artificial Intelligence is in a Crisis

“I would trust a fully autonomous vehicle.”

“I would trust a fully automatic technique to manage my finances.”
Artificial Intelligence is in a Crisis

“I would trust a fully autonomous vehicle.”

“I would trust a fully automatic technique to manage my finances.”

“I would trust a medical diagnosis that is made without a human involved.”
“I would trust a fully autonomous vehicle.”  

“I would trust a fully automatic technique to manage my finances.”

“I would trust a medical diagnosis that is made without a human involved.”

Bitkom research survey amongst German end users (2017)
Why don’t people trust AI?

AI systems make mistakes...

Microsoft silences its new A.I. bot Tay, after Twitter users teach it racism

Alexa, Nein! Police break into German man's house after music device 'held party on its own'

THIS APP IS HILARIOUS

I am an 11 year old gal. When using the app I get a few different numbers. A few were 12 and 15, (I felt a bit weird when I got that 15). BUT god I died of laughter when I got 88!
Why don’t people trust AI?

People also make mistakes...

My innocent client spent 25 years on death row. How long will it take to realize our system is broken?
Why don’t people trust AI?

AIs make the wrong mistakes.
Why don’t people trust AI?

stupid
Al’s make the wrong mistakes.
Why don’t people trust AI?

stupid
Als make the wrong mistakes.

To become trustworthy, learning system should behave more human-like:

- **robustness** to unexpected or even adversarial conditions,

  “Intelligent people are hard to fool.”
Why don’t people trust AI?

stupid
AIs make the wrong mistakes.

To become trustworthy, learning system should behave more human-like:

- **robustness** to unexpected or even adversarial conditions,
  
  “Intelligent people are hard to fool.”

- **introspection**, to be aware of its own performance, including failures,
  
  “Intelligent people are willing to admit when they are wrong or don’t know something.”
Why don’t people trust AI?

stupid
Als make the wrong mistakes.

To become trustworthy, learning system should behave more human-like:

- **robustness** to unexpected or even adversarial conditions,
  “Intelligent people are hard to fool.”

- **introspection**, to be aware of its own performance, including failures,
  “Intelligent people are willing to admit when they are wrong or don’t know something.”

- **adaptivity** to new situations or goals,
  “Intelligent people learn from their mistakes and don’t repeat them.”
Why don’t people trust AI?

stupid
AIs make the wrong mistakes.

To become trustworthy, learning system should behave more human-like:

- **robustness** to unexpected or even adversarial conditions,
  
  "Intelligent people are hard to fool."

- **introspection**, to be aware of its own performance, including failures,
  
  "Intelligent people are willing to admit when they are wrong or don’t know something."

- **adaptivity** to new situations or goals,
  
  "Intelligent people learn from their mistakes and don’t repeat them."

- **transparency, explainability** and **fairness** of the decision process.
  
  "I don’t trust a person who is prejudiced against me."

Excursion:
How do AI systems work?
Why are AI systems not trustworthy?

Today’s AI is based on **machine learning**, not traditional **software development**.

Traditional software development:
- target behavior described by a formal specification

(Supervised) Machine Learning:
- target behavior described by exemplary data
Software development: implement specifications

Example: write a subroutine that sorts a list

step 1) define formal specifications

1. “The output list should be a permutation of the input list.”

2. “The values in the output list should be in numerically increasing order.”

step 2) write code (manually), result: a subroutine $f : \{\text{lists}\} \rightarrow \{\text{lists}\}$

subroutine bubbleSort(list A)
   for (n=size(A); n>1; n=n-1) {
      for (i=0; i<n-1; i=i+1) {
         if (A[i] > A[i+1]) {
            swap(A[i], A[i+1])
         }
      }
   }

step 3) make sure that $f$ does what it is supposed to do:

- a) testing:
  feed input lists to the routine and check if outputs are indeed sorted

- b) formal analysis and verification:
  prove mathematically that the specifications are fulfilled for any possible input
Machine learning: solve a task for which we only have an informal description

Example: **building a system that analyses texts for their sentiment.**

Are these movie reviews positive or negative?

“*This short movie is the best.*”

vs.

“The *best thing about this movie is how short it is.*”

Humans are quite good at this task, but we cannot formally describe how to do it.
Machine learning: solve a task for which we only have an informal description

Example: building a system that analyses texts for their sentiment.

step 0) install an existing text classification library

step 1) create a dataset
- example inputs: reviews of the type you care about
- target outputs: positive/negative labels assigned by a (human) expert

step 2) run the library’s training routine
- result: a subroutine \( f : \{\text{texts}\} \rightarrow \{\text{positive, negative}\} \)
step 3) how to make sure that $f$ does what it is supposed to do?

Can we test it? Yes, though not fully automatically:
→ collect more human-written reviews,
→ ask human expert to provide correct labels,
→ compare the classifier outputs to correct labels.

Can we verify it? No, we don’t have any formal specifications.

Can we analyze it? (Usually) No, it’s too complex, especially for deep learning.
“Program testing can be used to show the presence of bugs, but never to show their absence!”

Edsger Dijkstra, 1970
Excursion: How Deep Learning Works
Excursion: Deep Learning

(artificial) neural network model

- many ‘neurons’, arranged in many layers
- each neuron has many inputs $i_1, i_2, \ldots, i_k$ and one output $o$
- each neuron computes a simple function

$$o = \max\{0, w_0 + \sum_{j=1}^{k} w_j i_j\}$$

- for each neuron, the connection weights $w_1, \ldots, w_k$ are trainable

Overall: highly-nonlinear function, millions of trainable weights
Excursion: Deep Learning

Initialization:
- assign random values to connection weights
Excursion: Deep Learning

Training phase:
- present a training example to network
- compute its prediction
- check if prediction is correct
- if not, adjust connection weights
- repeat, until no more training examples

Repeat for several hundred epochs.
Excursion: Deep Learning

Training phase:
- present a training example to network
- compute its prediction
- check if prediction is correct
- if not, adjust connection weights
- repeat, until no more training examples

Repeat for several hundred epochs.
Excursion: Deep Learning

**Training phase:**

1. Present a training example to network
2. Compute its prediction
3. Check if prediction is correct
4. If not, adjust connection weights
5. Repeat, until no more training examples

Repeat for several hundred epochs.
Excursion: Deep Learning

Training phase:
- present a training example to network
- **compute its prediction**
- check if prediction is correct
- if not, adjust connection weights
- repeat, until no more training examples

Repeat for several hundred epochs.
Excursion: Deep Learning

Training phase:
- present a training example to network
- **compute its prediction**
- check if prediction is correct
- if not, adjust connection weights
- repeat, until no more training examples

Repeat for several hundred epochs.
Excursion: Deep Learning

Training phase:
- present a training example to network
- compute its prediction
- check if prediction is correct
- if not, adjust connection weights
- repeat, until no more training examples

Repeat for several hundred epochs.
Excursion: Deep Learning

Training phase:
- present a training example to network
- **compute its prediction**
- check if prediction is correct
- if not, adjust connection weights
- repeat, until no more training examples

Repeat for several hundred epochs.
Excursion: Deep Learning

Training phase:
- present a training example to network
- compute its prediction
- check if prediction is correct
- if not, adjust connection weights
- repeat, until no more training examples

Repeat for several hundred epochs.
Excursion: Deep Learning

Training phase:
- present a training example to network
- compute its prediction
- check if prediction is correct
- if not, adjust connection weights
- repeat, until no more training examples

Repeat for several hundred epochs.
Excursion: Deep Learning

Training phase:
- present a training example to network
- compute its prediction
- check if prediction is correct
- if not, adjust connection weights
- repeat, until no more training examples

Repeat for several hundred epochs.
Excursion: Deep Learning

Training phase:
- present a training example to network
- compute its prediction
- **check if prediction is correct**
- if not, adjust connection weights
- repeat, until no more training examples

Repeat for several hundred epochs.
Excursion: Deep Learning

Training phase:
- present a training example to network
- compute its prediction
- check if prediction is correct
- if not, adjust connection weights
- repeat, until no more training examples

Repeat for several hundred epochs.
Excursion: Deep Learning

Training phase:
- present a training example to network
- compute its prediction
- check if prediction is correct
- if not, adjust connection weights
- repeat, until no more training examples

Repeat for several hundred epochs.
Excursion: Deep Learning

Training phase:
- present a training example to network
- compute its prediction
- check if prediction is correct
- if not, adjust connection weights
- repeat, until no more training examples

Repeat for several hundred epochs.
Excursion: Deep Learning

Training phase:
- present a training example to network
- compute its prediction
- check if prediction is correct
- if not, adjust connection weights
- repeat, until no more training examples

Repeat for several hundred epochs.
Excursion: Deep Learning

Training phase:
- present a training example to network
- compute its prediction
- check if prediction is correct
- if not, adjust connection weights
- repeat, until no more training examples

Repeat for several hundred epochs.
Excursion: Deep Learning

Training phase:
- present a training example to network
- compute its prediction
- check if prediction is correct
- if not, adjust connection weights
- repeat, until no more training examples

Repeat for several hundred epochs.
Excursion: Deep Learning

Training phase:
- present a training example to network
- compute its prediction
- check if prediction is correct
- if not, adjust connection weights
- repeat, until no more training examples

Repeat for several hundred epochs.
Excursion: Deep Learning

Training phase:
- present a training example to network
- compute its prediction
- check if prediction is correct
- if not, adjust connection weights
- repeat, until no more training examples

Repeat for several hundred epochs.
Excursion: Deep Learning

Training phase:
- present a training example to network
- compute its prediction
- check if prediction is correct
- if not, adjust connection weights
- repeat, until no more training examples

Repeat for several hundred epochs.
Excursion: Deep Learning

Deployment phase:
- connection weights remain fixed
- network can make predictions
- no more learning
Excursion: Deep Learning

Deployment phase:
- connection weights remain fixed
- network can make predictions
- no more learning
Excursion: Deep Learning

Deployment phase:
- connection weights remain fixed
- network can make predictions
- no more learning
Excursion: Deep Learning

Deployment phase:
- connection weights remain fixed
- network can make predictions
- no more learning
Excursion: Deep Learning

Deployment phase:
- connection weights remain fixed
- network can make predictions
- no more learning
Excursion: Deep Learning

Deployment phase:
- connection weights remain fixed
- network can make predictions
- no more learning

We can follow the numeric operations step by step, but we don’t truly understand them.
To become trustworthy, learning system should behave more human-like:

- **robustness** to unexpected or even adversarial conditions,
- **introspection**, to be aware of its own performance, including failures,
- **adaptivity** to new situations or goals,
- **transparency**, **explainability** and **fairness** of the decision process.
Robustness to unexpected or even adversarial conditions

**Statistical learning theory**: for any function $f$:

$$\mathbb{E}[\ell_f(x)] \leq \frac{1}{n} \sum_{i=1}^{n} \ell_f(x_i) + \frac{\text{const.}}{\sqrt{n}}$$

expected error on future data
observed error on training data
sampling uncertainty

This holds under the assumptions that:
- future data will be random samples from some probability distribution
- training data is a set of samples from this same probability distribution

Problem:
- In real life, these assumptions are often violated.
The training data rarely perfectly reflects future data:

- data collection bias
- annotation bias
- static-world bias

Example: domain shift

truck? or lorry?
Example: adversarial examples

Image classification: recognize 1000 object categories in natural images

- state-of-the-art deep (convolutional) neural network
- trained on 1.2 millions images
- tested on 100,000 additional images:
  - predicts correct category on more than 80%
  - correct category in top-5 predictions in more than 95%
Example: adversarial examples

Image classification: recognize 1000 object categories in natural images

- state-of-the-art deep (convolutional) neural network
- trained on 1.2 millions images
- tested on 100,000 additional images:
  - predicts correct category on more than 80%
  - correct category in top-5 predictions in more than 95%
Example: adversarial examples

Image classification: recognize 1000 object categories in natural images

- state-of-the-art deep (convolutional) neural network
- trained on 1.2 millions images
- tested on 100,000 additional images:
  - predicts correct category on more than 80%
  - correct category in top-5 predictions in more than 95%

Example 1: zebra (99% confidence)
Example: adversarial examples

Image classification: recognize 1000 object categories in natural images

- state-of-the-art deep (convolutional) neural network
- trained on 1.2 millions images
- tested on 100,000 additional images:
  - predicts correct category on more than 80%
  - correct category in top-5 predictions in more than 95%
Example: adversarial examples

Image classification: recognize 1000 object categories in natural images

- state-of-the-art deep (convolutional) neural network
- trained on 1.2 millions images
- tested on 100,000 additional images:
  - predicts correct category on more than 80%
  - correct category in top-5 predictions in more than 95%

Example 2:
Example: adversarial examples

Image classification: recognize 1000 object categories in natural images

- state-of-the-art deep (convolutional) neural network
- trained on 1.2 millions images
- tested on 100,000 additional images:
  - predicts correct category on more than 80%
  - correct category in top-5 predictions in more than 95%

Example 2:

![Zebra](image.jpg) toaster (97% confidence)
Example: adversarial examples

Example 1: natural image, downloaded from the Internet
Example 2: artificially modified to confuse the network (= not a random sample)
Adversarial examples are like optical illusions for neural networks.
Overview

To become trustworthy, learning system should behave more human-like:

- **robustness** to unexpected or even adversarial conditions,

- **introspection**, to be aware of its own performance, including failures,

- **adaptivity** to new situations or goals,

- **transparency**, **explainability** and **fairness** of the decision process.
**Introspection**, to be aware of its own performance, including failures

<table>
<thead>
<tr>
<th>input image</th>
<th>ideal output</th>
<th>ski</th>
<th>ski (rotated camera)</th>
<th>nothing (just black)</th>
<th>nothing (random noise)</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="input image" /></td>
<td><img src="image2.png" alt="ideal output" /></td>
<td><img src="image3.png" alt="ski" /></td>
<td><img src="image4.png" alt="ski (rotated camera)" /></td>
<td><img src="image5.png" alt="nothing (just black)" /></td>
<td><img src="image6.png" alt="nothing (random noise)" /></td>
</tr>
</tbody>
</table>
**Introspection**, to be aware of its own performance, including failures

<table>
<thead>
<tr>
<th>input image</th>
<th>ideal output</th>
<th>prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="image" /></td>
<td>ski</td>
<td>ski</td>
</tr>
<tr>
<td><img src="image2.png" alt="image" /></td>
<td>ski (rotated camera)</td>
<td>paintbrush</td>
</tr>
<tr>
<td><img src="image3.png" alt="image" /></td>
<td>nothing (just black)</td>
<td>web site</td>
</tr>
<tr>
<td><img src="image4.png" alt="image" /></td>
<td>nothing (random noise)</td>
<td>tennis ball</td>
</tr>
</tbody>
</table>

Currently used classifiers lack **introspection**:
- they will always predict one of the fixed set of labels they trained for.
- they are unable to detect situations for which they were not trained.
- they are unable to say "*I don’t know.*"
To become trustworthy, learning system should behave more human-like:

- **robustness** to unexpected or even adversarial conditions,
- **introspection**, to be aware of its own performance, including failures,
- **adaptivity** to new situations or goals,
- **transparency, explainability** and **fairness** of the decision process.
Adaptivity to new situations or goals

Image classification: recognize 1000 object categories in natural images

- state-of-the-art deep (convolutional) neural network
- trained on 1.2 millions images, collected from the Internet in 2012

Example:

- pencil sharpener
Adaptivity to new situations or goals

Image classification: recognize 1000 object categories in natural images

- state-of-the-art deep (convolutional) neural network
- trained on 1.2 millions images, collected from the Internet in 2012

Example:

![Pencil sharpener/fidget spinner](image)
Adaptivity to new situations or goals

Image classification: recognize 1000 object categories in natural images

- state-of-the-art deep (convolutional) neural network
- trained on 1.2 millions images, collected from the Internet in 2012

Example:

After the training phase is over, networks are unable to learn from their mistakes.
To become trustworthy, learning system should behave more human-like:

- **robustness** to unexpected or even adversarial conditions,
- **introspection**, to be aware of its own performance, including failures,
- **adaptivity** to new situations or goals,
- **transparency**, **explainability** and **fairness** of the decision process.
Transparency, explainability and fairness of the decision process

Scenario 1:
- A postdoc applies for a job to me, but gets rejected.
- She asks “Why?”
- “Because you do not have enough high-quality publications.”

Scenario 2:
- A postdoc applies for a job, but gets rejected by a neural network.
- She asks “Why?”
- “Because the network’s output was a negative number.”

People don’t just want decisions, they want explanations.
Transparency, explainability and fairness of the decision process

Simple models
- e.g. naive Bayes, decision trees, ...
  - are (often) easy to explain
  - provide only limited accuracy

Complex models,
- e.g. deep neural networks
  - offer high classification accuracy
  - decisions are hard/impossible to explain

#papers ≥ 3

accept

reject

text document

yes

no

#papers ≥ 3

1 2 3 4 5

text document
Transparency, explainability and fairness of the decision process

Scenario 3:
- A postdoc applies for a job to me, but gets rejected.
- She asks “Is that because I am a woman?”.
- “No, it’s because you do not have enough high-quality publications.”

Scenario 4:
- A postdoc applies for a job, but gets rejected by a neural network.
- She asks “Is that because I am a woman?”
- “Maybe. We really don’t know.”

Decisions should be fair and unbiased.
Transparency, explainability and fairness of the decision process

Example: Google Translate has a gender bias

<table>
<thead>
<tr>
<th>English – detected</th>
<th>Turkish</th>
</tr>
</thead>
<tbody>
<tr>
<td>She's smart.</td>
<td>O akıllı.</td>
</tr>
<tr>
<td>He's beautiful.</td>
<td>O çok güzel.</td>
</tr>
<tr>
<td>She is a doctor.</td>
<td>O bir doktor.</td>
</tr>
<tr>
<td>He is a nurse.</td>
<td>O bir hemşire.</td>
</tr>
</tbody>
</table>
Transparency, explainability and fairness of the decision process

Example: Google Translate has a **gender bias**

English – detected  

She's smart.  
He's beautiful.  
She is a doctor.  
He is a nurse.

Turkish  

O akıllı.  
O çok güzel.  
O bir doktor.  
O bir hemşire.

Turkish  

O akıllı.  
O çok güzel.  
O bir doktor.  
O bir hemşire.

English  

He's smart.  
She's beautiful.  
He is a doctor.  
She is a nurse.
Towards trustworthy machine learning...
Introspection, to be aware of its own performance, including failures

KS(conf): a Light-Weight Test if a ConvNet Operates Outside of its Specifications.

arXiv:1804.04171 [stat.ML]
Introspection, to be aware of its own performance, including failures

Situation: a user wants to run a trained predicted model for a long time.

Can we tell automatically...

- if the model makes correct predictions or not?
  → too hard
- if the input data is of the same type as what the model was trained for?
  → if not, chances are high that predictions are unreliable → warn the user

Statistical “two sample” test:

- given two data sets, do they both come from the same data distribution?
**Introspection**, to be aware of its own performance, including failures

Our solution: **KS(conf)**

- compare **statistics of network outputs** (confidence scores) instead of inputs
  → overcomes the curse-of-dimensionality

- work on **batches** (groups of images) instead of single images
  → allows us to control the false positive rate

- use **Kolmogorov-Smirnov (KS)** test to compare score distributions
  → efficient, distribution-free, well-understood null distribution
**Introspection**, to be aware of its own performance, including failures

Illustration: five different 1000-class object detection networks

Score distribution on original data:

Score distribution on images of other classes (AwA2 dataset):

→ difference can be detected reliably
**Introspection**, to be aware of its own performance, including failures

Score distribution on original data:

Score distribution on rotated images:

→ difference can be detected reliably
Introspection, to be aware of its own performance, including failures

Score distribution on original data:

Score distribution on images with dead pixels:

→ difference can be detected reliably
**Introspection**, to be aware of its own performance, including failures

Score distribution on original data:

Score distribution on images with dead pixels:

10% dead pixels

100% dead pixels
Introspection, to be aware of its own performance, including failures

Quantitative results (avg. detection rate):

<table>
<thead>
<tr>
<th></th>
<th>KS (conf)</th>
<th>mean</th>
<th>logmean</th>
<th>z</th>
<th>log-z</th>
<th>sym.mean</th>
<th>sym.logmean</th>
<th>sym.z</th>
<th>sym.log-z</th>
</tr>
</thead>
<tbody>
<tr>
<td>AwA2-bat</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>AwA2-blue whale</td>
<td>1.00</td>
<td>0.20</td>
<td>0.00</td>
<td>0.20</td>
<td>0.00</td>
<td>0.80</td>
<td>0.00</td>
<td>0.80</td>
<td>0.62</td>
</tr>
<tr>
<td>AwA2-bobcat</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>AwA2-dolphin</td>
<td>1.00</td>
<td>1.00</td>
<td>0.79</td>
<td>1.00</td>
<td>0.79</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.70</td>
</tr>
<tr>
<td>AwA2-giraffe</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>AwA2-horse</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>AwA2-rat</td>
<td>1.00</td>
<td>1.00</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>1.00</td>
<td>0.20</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>AwA2-seal</td>
<td>1.00</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>1.00</td>
<td>0.20</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>AwA2-sheep</td>
<td>1.00</td>
<td>0.01</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.65</td>
<td>0.65</td>
<td>0.65</td>
<td>0.82</td>
</tr>
<tr>
<td>AwA2-walrus</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>DAVIS</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
</tbody>
</table>
Adaptivity to new situations or goals

iCaRL: Incremental Classifier and Representation Learning

CVPR 2017
arXiv:1611.07725 [cs.CV]
Adaptivity to new situations or goals

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
Adaptivity to new situations or goals

iCaRL (incremental classifier and representation learning):
- train **fixed-size deep network** with ordinary BackProp
- keep a small **set of exemplars** from all classes seen so far
- classify using ‘**nearest-mean-of-exemplars’** rule instead of network outputs

Results:

![Graphs](image-url) Multi-class accuracies over 10 repeats (average and standard deviation) for class-incremental training on CIFAR-100
Adaptivity to new situations or goals

iCaRL (incremental classifier and representation learning):
- train fixed-size deep network with ordinary BackProp
- keep a small set of exemplars from all classes seen so far
- classify using ‘nearest-mean-of-exemplars’ rule instead of network outputs

Results:

Confusion matrices of different method on CIFAR-100 after training for 100 classes with 10 classes per batch
Towards Practical Conditional Risk Minimization

arXiv:1801.00507 [stat.ML]

Alex Zimin
(IST Austria)
Transparency, explainability and fairness

Observation: a single simple model is usually not sufficient for high accuracy

Illustrative example: sentiment analysis with per-word scores

- each word gets a positive, neutral or negative score.
- the overall score of a review is the average of word scores.

This book is awesome.
0 0 0 +1 → positive review.

I hate this terrible movie.
0 -1 0 -1 0 → negative review.

Efficient and explainable, but doesn’t always work.
Transparency, explainability and fairness

For some important words no single positive/negative score makes sense.

This knife is really sharp.
0 0 0 0 0 ? → should be positive

The crib had sharp edges.
0 0 0 ? 0 0 → should be negative

A single word-score model cannot reflect both.

Idea: learn several models, and switch between them based on context.
Transparency, explainability and fairness

**Theorem 1.** If MACRO is run with ERM as a subroutine, then we have for any $k, m \geq 1$, $\alpha \in [0, 1]$ and $\beta \in [0, \alpha/4]$

\[
\mathbb{P} \left[ R_n(h_n) - \inf_{h \in \mathcal{H}} R_n(h) > \alpha + 2\varepsilon \right]
\leq \frac{2kN_{\infty}(\mathcal{L}(\mathcal{H}), \beta, n)}{(\alpha - 4\beta)^2} e^{-\frac{1}{2}m(\alpha - 4\beta)^2} + \mathbb{P} \left[ E_{k,m}^c \right].
\]  

(10)

**Theorem 2.** For a convex loss $\ell$, if the subroutines of MACRO use an averaging for online-to-batch conversion, we have for any $\alpha \in [0, 1]$ and $\beta \in [0, \alpha/8]$

\[
\mathbb{P} \left[ R_n(h_n) - \inf_{h \in \mathcal{H}} R_n(h) > \alpha + W_{I_{n,n}}/s_{I_{n,n}} + 4\varepsilon \right]
\leq \frac{4kN_{\infty}(\mathcal{L}(\mathcal{H}), \beta, n)}{(\alpha/2 - 4\beta)^2} e^{-\frac{1}{2}m(\alpha/2 - 4\beta)^2} + \mathbb{P} \left[ E_{k,m}^c \right].
\]  

(13)

MACRO:

**Initialization:** $T \leftarrow \emptyset$, $N \leftarrow 0$

At any time point $t = 1, 2, \ldots$:

- **choose** as active hypothesis the output of the closest $\varepsilon$-close subroutine or a newly started one
  - identify all $\varepsilon$-close subroutines
    
    \[ J = \{ j \in T : M_{t,j} \leq \varepsilon \} \]
  - if $J = \emptyset$: create a new subroutine, $S_{N+1}$,
    
    \[ J = \{ N+1 \}, \quad T \leftarrow T \cup \{ N+1 \}, \quad N \leftarrow N+1 \]
  - make the output of the closest subroutine active
    
    \[ h_a \leftarrow \text{output}(S_j) \quad \text{for } j = \arg\min_{j \in T} M_{t,j} \]
- **output** the currently active hypothesis, $h_t \leftarrow h_a$
- **observe** the next value of the process, $z_t$
- **update** all $\varepsilon$-close subroutines:
    
    \[ S_j \leftarrow \text{update}(S_j, z_t) \quad \text{for all } j \in J \]
Summary

- Artificial Intelligence has great potential.
- People do not trust AI for important tasks.
- To make AI trustworthy, systems need more human-like qualities (especially in their mistakes):
  - robustness
  - introspection
  - adaptivity
  - transparency, explainability and fairness
- Lots of open research questions...

Funding sources:

Group members and alumni: