Trustworthy Machine Learning

Christoph Lampert

Renningen, May 22, 2018

IST Austria (Institute of Science and Technology Austria)

Questions? chl@ist.ac.at

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

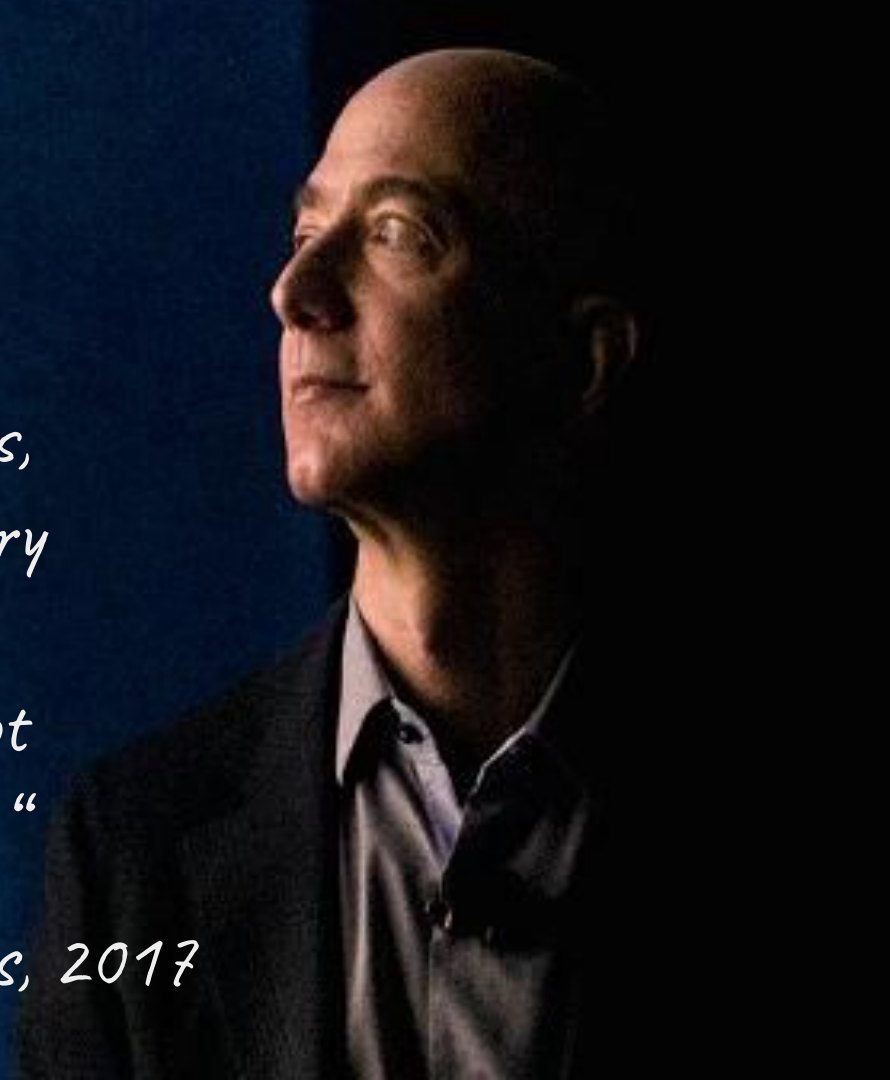
Focus on Basic Research

- 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

Artificial Intelligence is Flourishing



"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

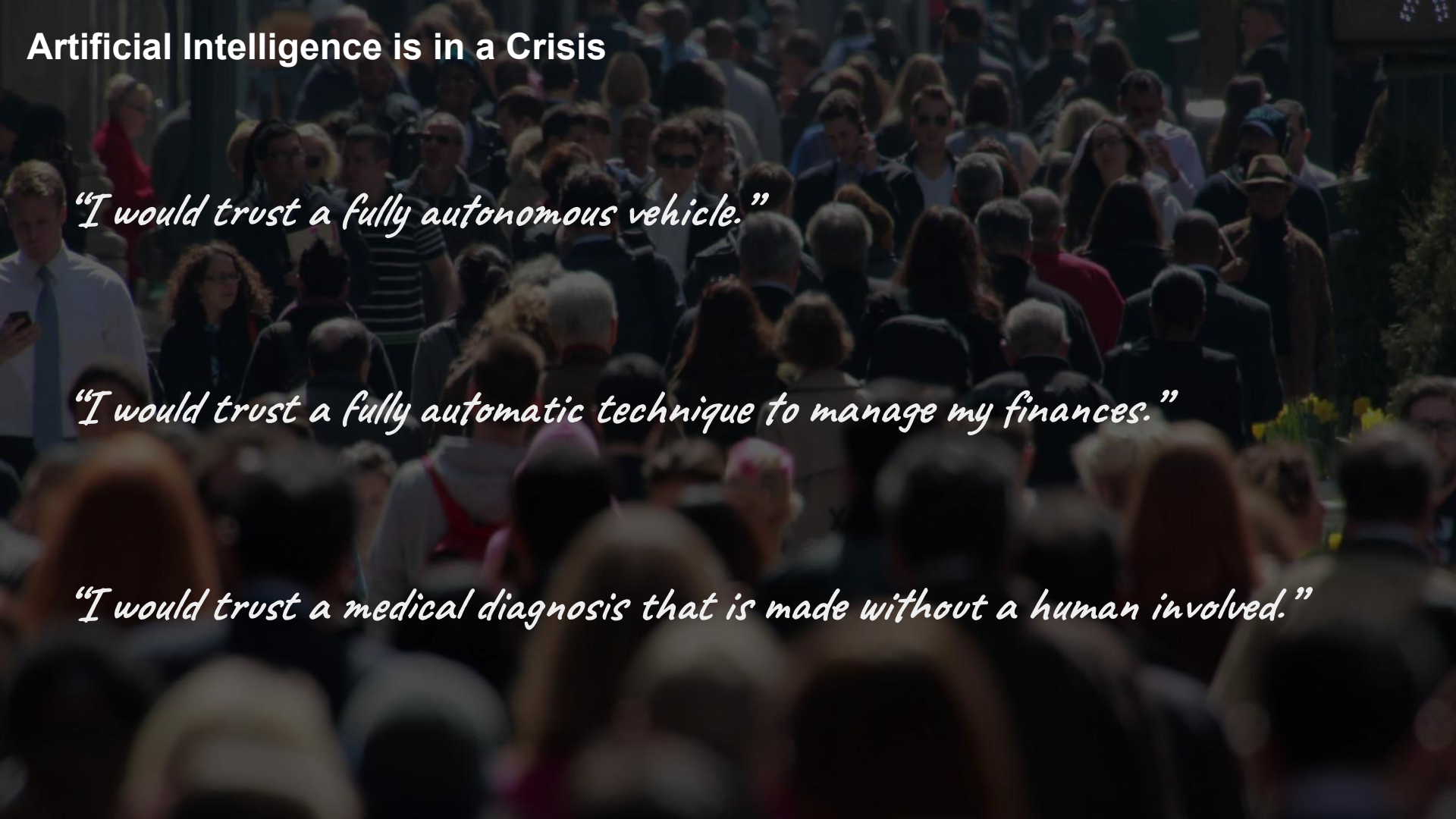
Artificial Intelligence is in a Crisis

Artificial Intelligence is in a Crisis

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."

Artificial Intelligence is in a Crisis

"I would trust a fully autonomous vehicle." 41%

"I would trust a fully automatic technique to manage my finances." 16%

"I would trust a medical diagnosis that is made without a human involved." 7%

Bitkom research survey amongst German end users (2017)

Why don’t people trust AI?

AI systems make mistakes...

TC

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

The Tele

News

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

of 5 448 Ratings

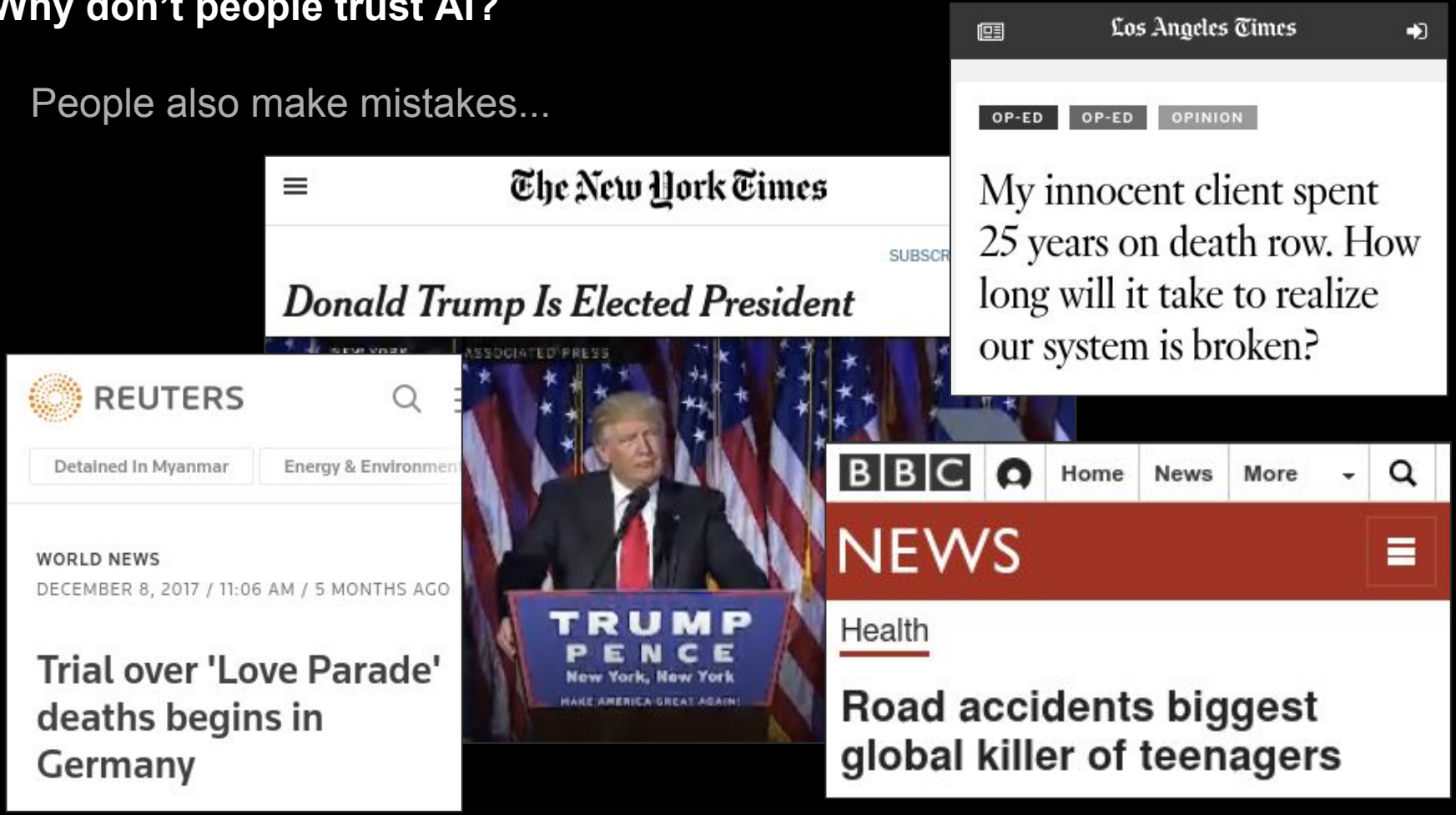
THIS APP IS HILARIOUS Jan 22, 2018

★★★★★ I.F. puppy

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).

Why don’t people trust AI?

People also make mistakes...



Why don’t people trust AI?

AIs make the wrong mistakes.

Why don’t people trust AI?

stupid

AIs make ~~the wrong~~ mistakes.

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.”

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
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.”

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

```
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])
      }
    }
  }
```

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 $f : \{\text{lists}\} \rightarrow \{\text{lists}\}$

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 $f : \{\text{texts}\} \to \{\text{positive}, \text{negative}\}$

Machine learning: solve a task for which we only have an informal description

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

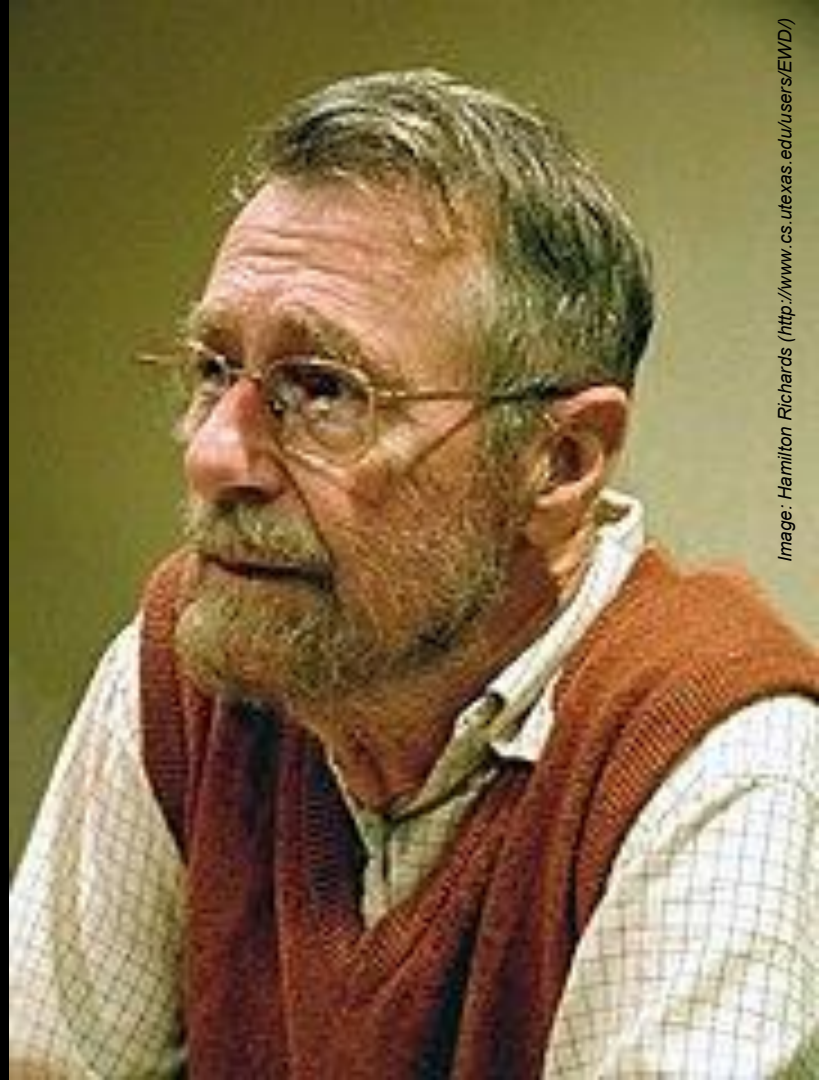


Image: Hamilton Richards (http://www.cs.utexas.edu/users/EWD/)

Excursion:
How Deep Learning Works

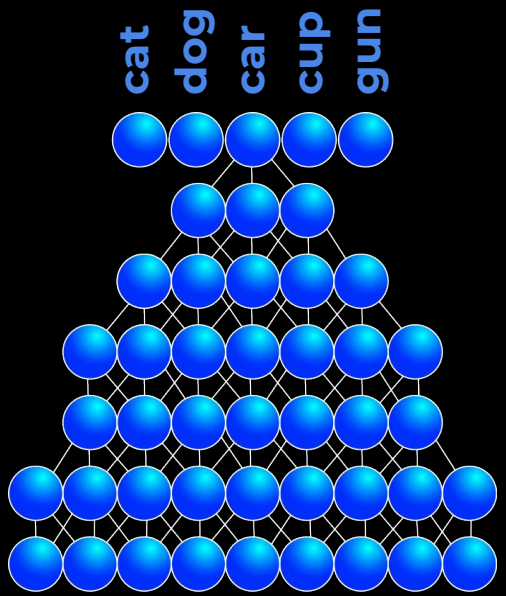
Excursion: Deep Learning

(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

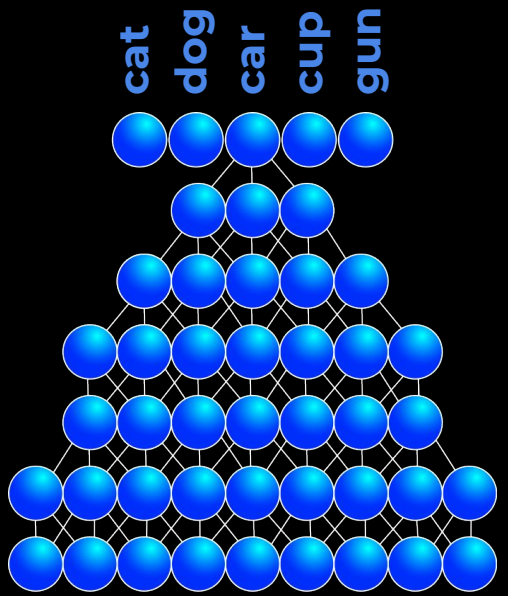


Overall: highly-nonlinear function, millions of trainable weights

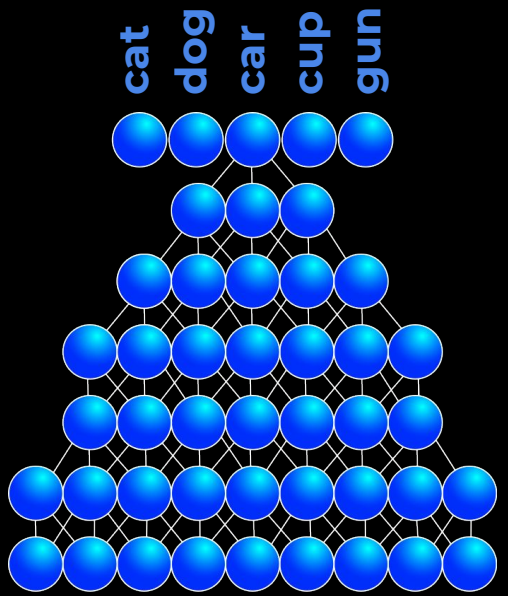
Excursion: Deep Learning

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

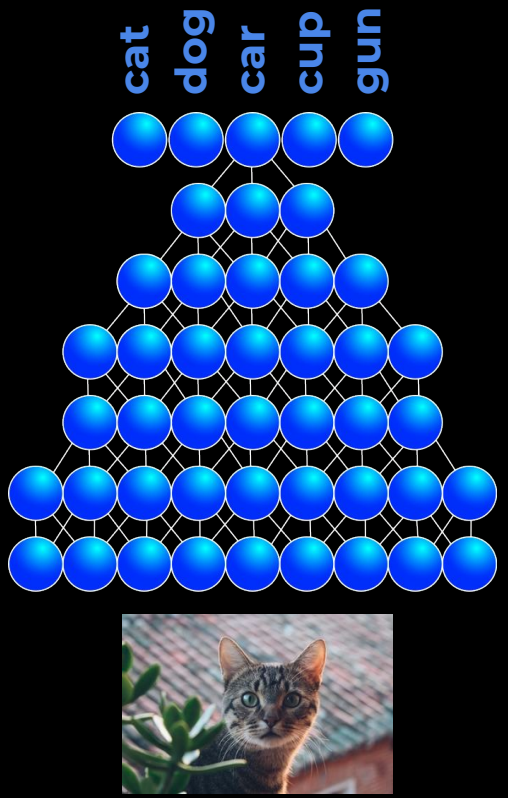
Repeat for several hundred epochs.

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.

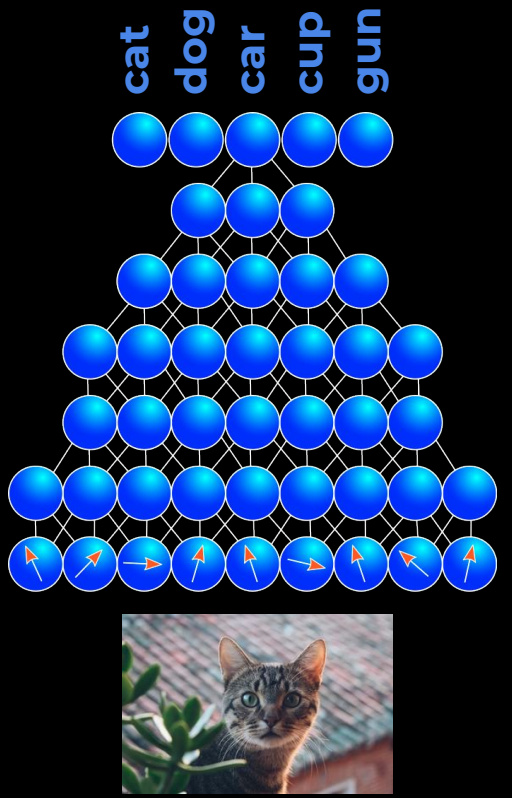


Excursion: Deep Learning

Training phase:

- 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.

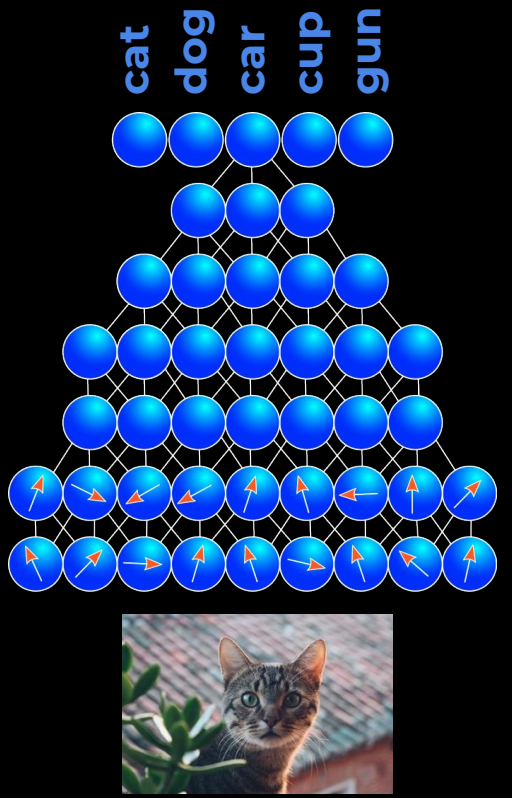


Excursion: Deep Learning

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.

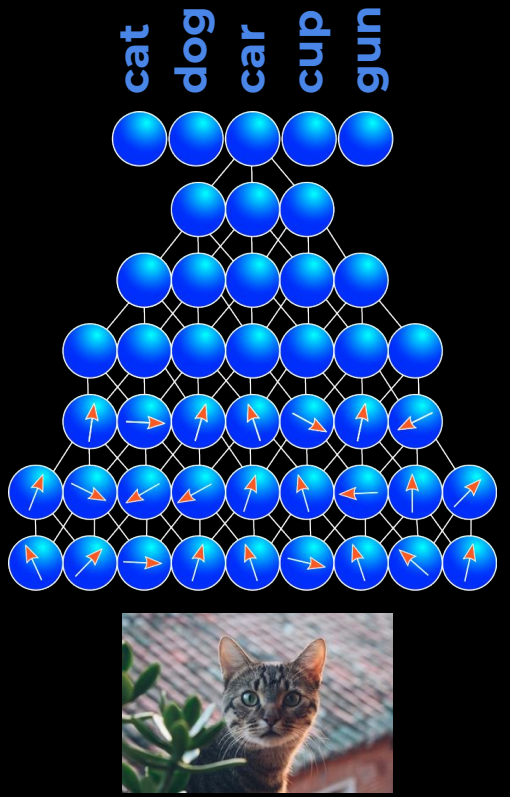


Excursion: Deep Learning

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.

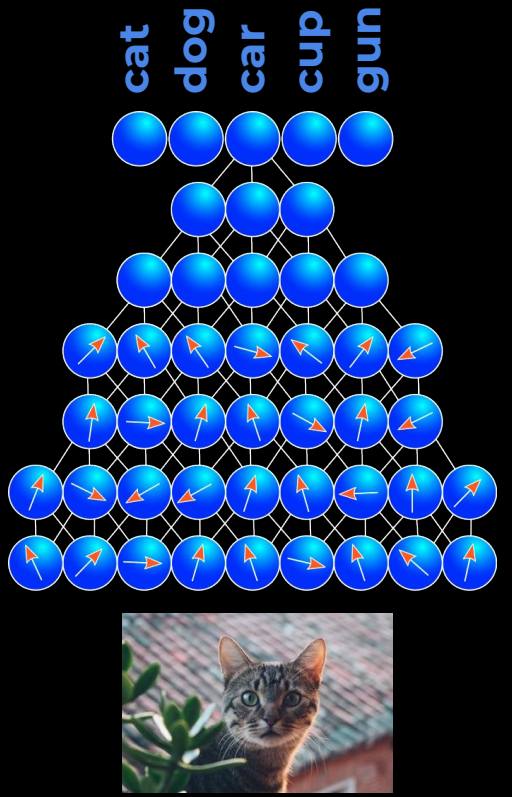


Excursion: Deep Learning

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.

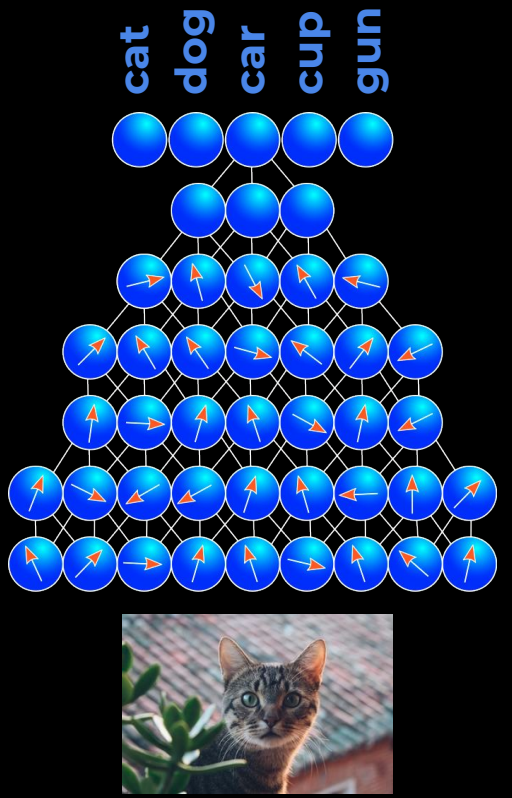


Excursion: Deep Learning

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.

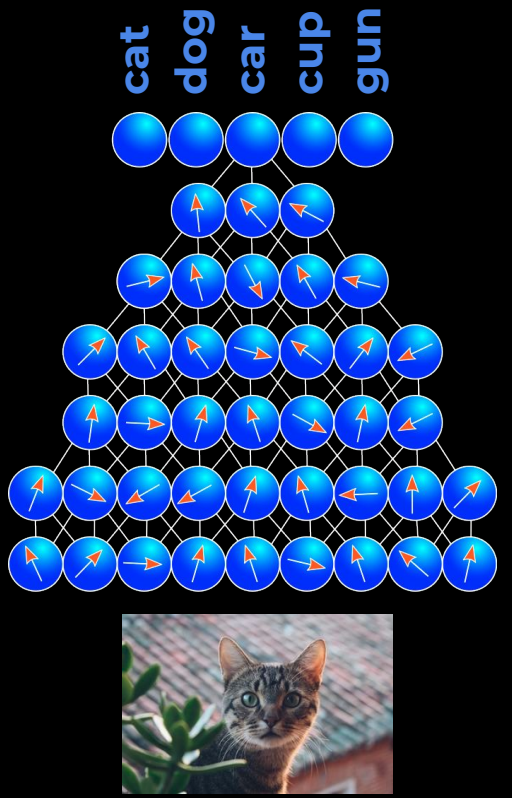


Excursion: Deep Learning

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.

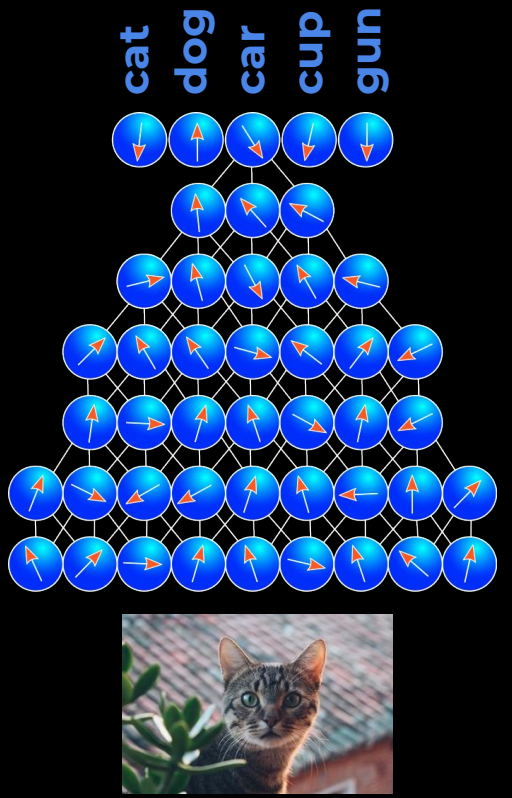


Excursion: Deep Learning

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.

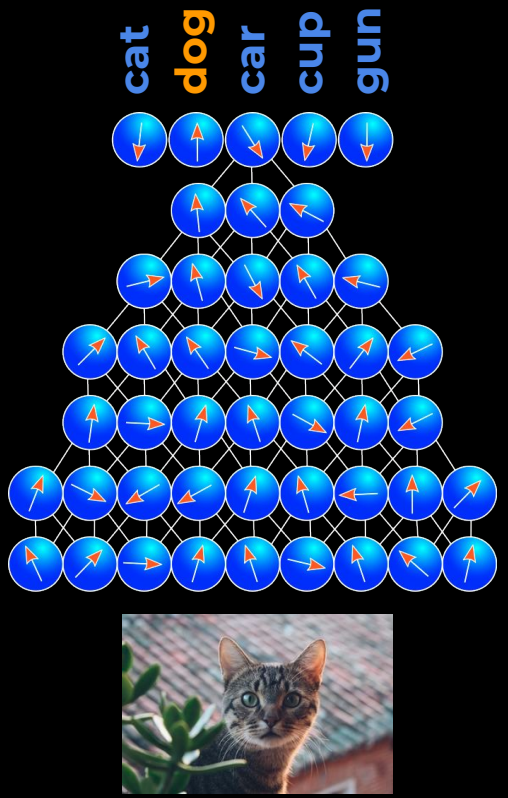


Excursion: Deep Learning

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.

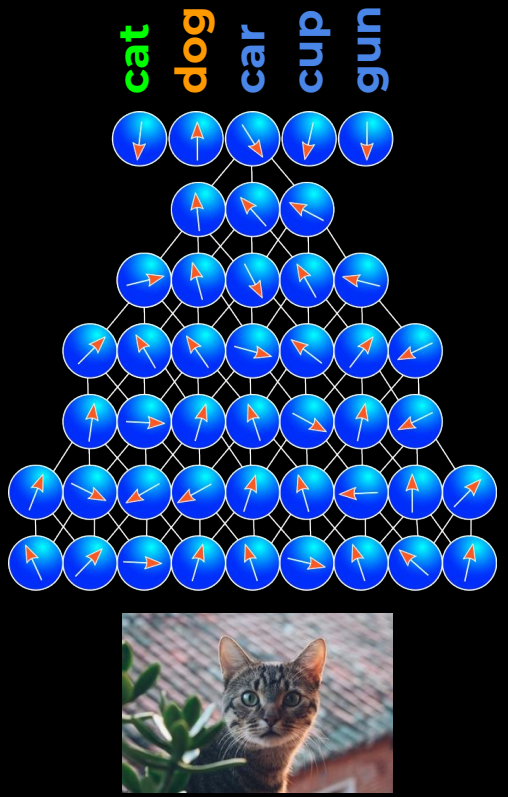


Excursion: Deep Learning

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.

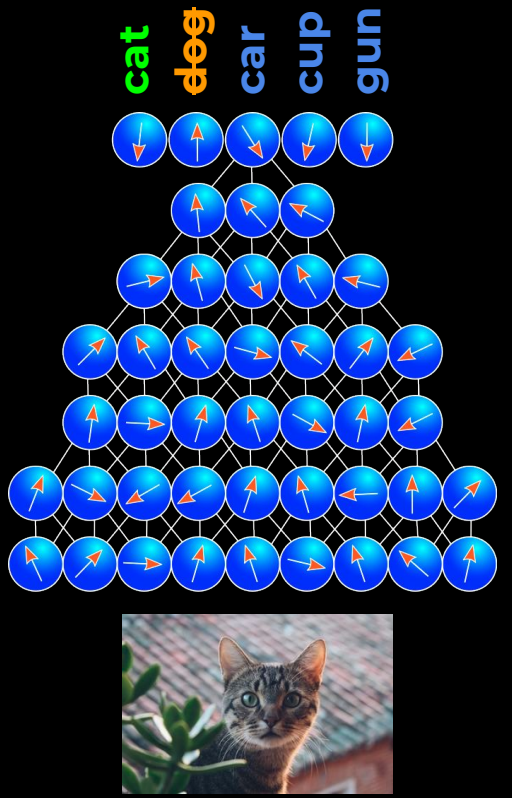


Excursion: Deep Learning

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.

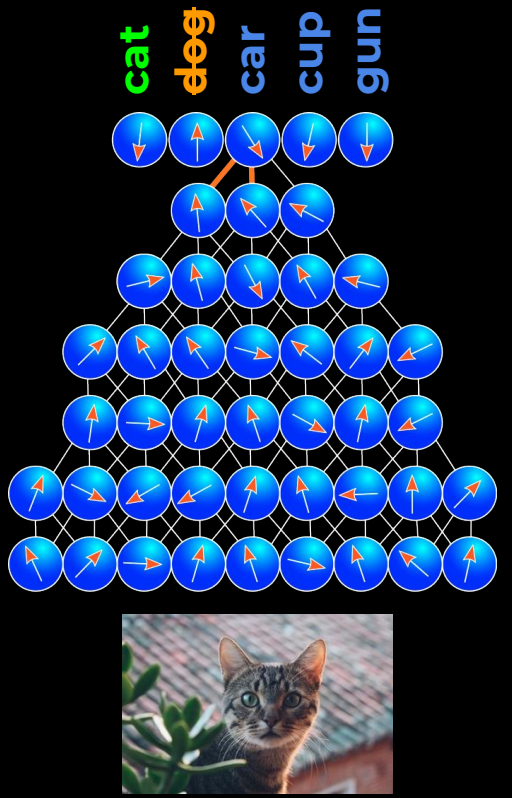


Excursion: Deep Learning

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.

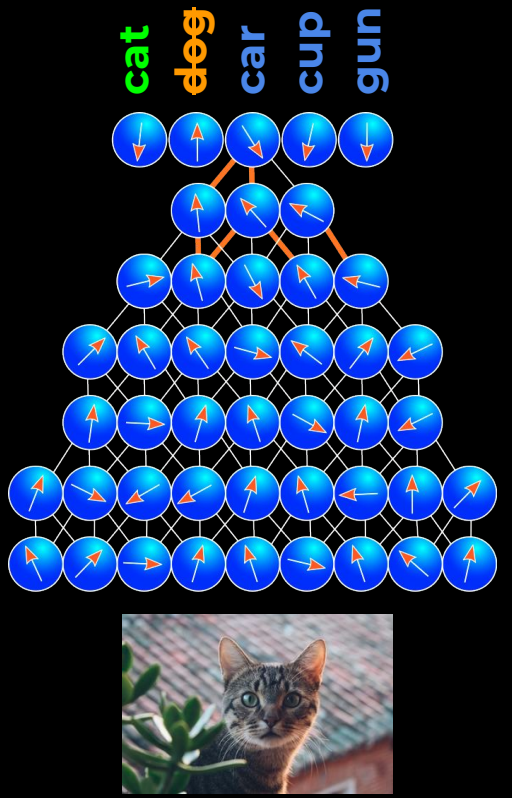


Excursion: Deep Learning

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.

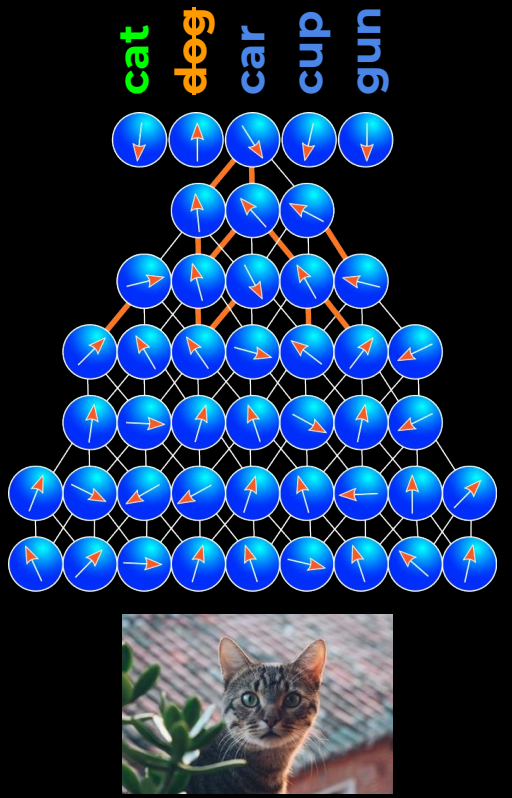


Excursion: Deep Learning

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.

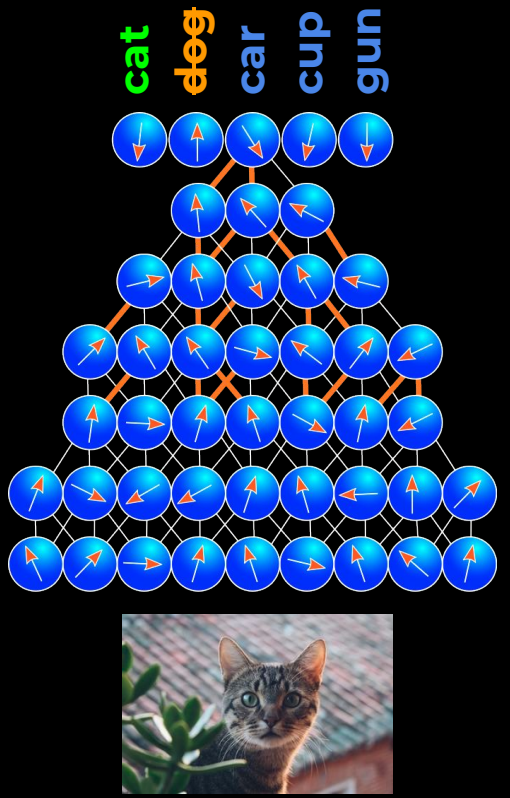


Excursion: Deep Learning

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.

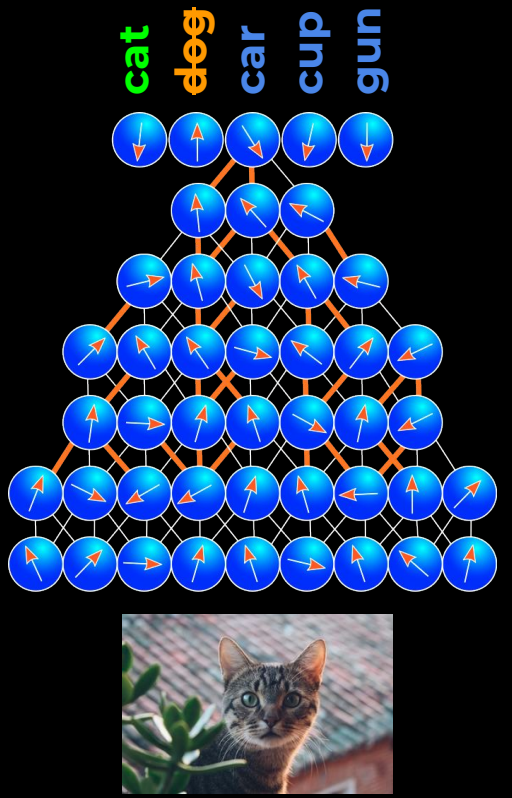


Excursion: Deep Learning

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.

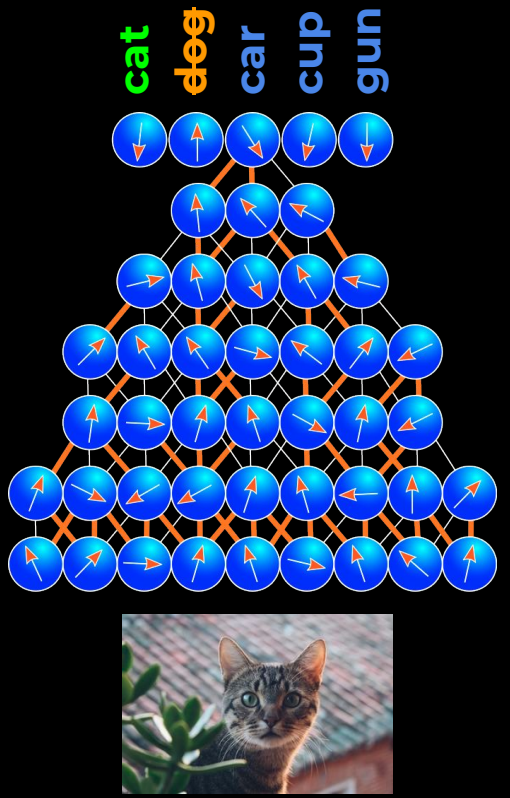


Excursion: Deep Learning

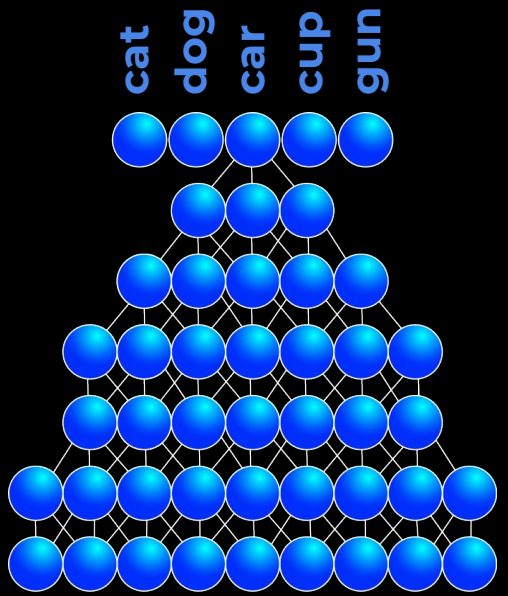
Training phase:

- 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

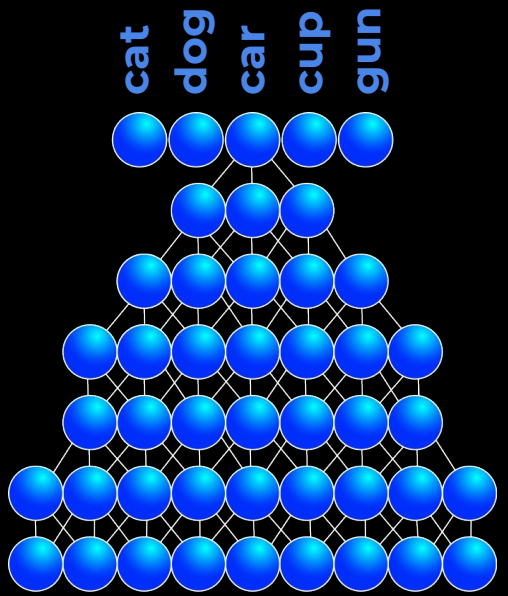


Training phase:

- 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



Training phase:

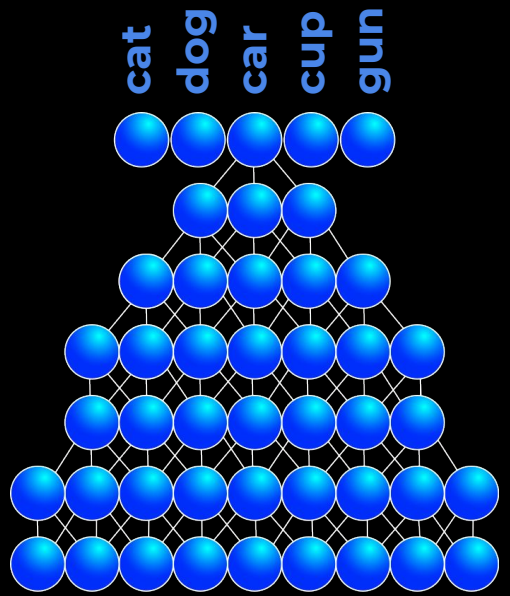
- present a training example to network
- compute its prediction
- check if prediction is correct
- if not, adjust connection weights
- repeat, until no more training examples

Repeat for several hundred epochs.

Excursion: Deep Learning

Deployment phase:

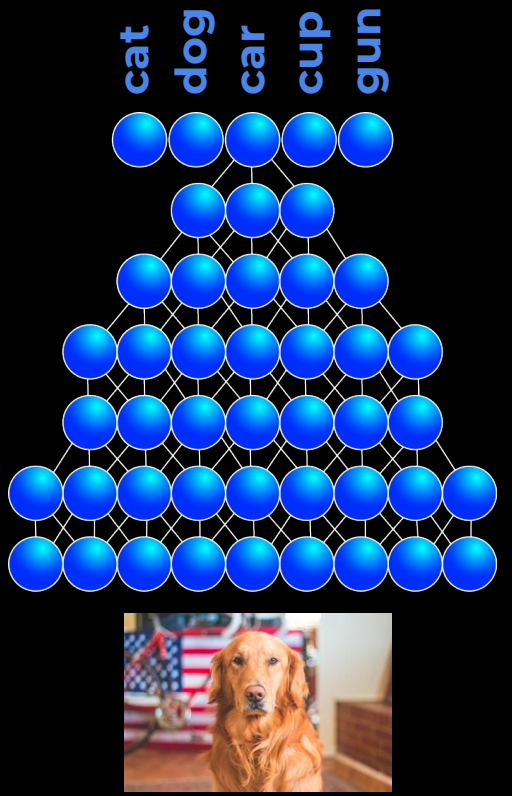
- connection weights remain fixed
- network can make predictions
- no more learning



Excursion: Deep Learning

Deployment phase:

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



Excursion: Deep Learning

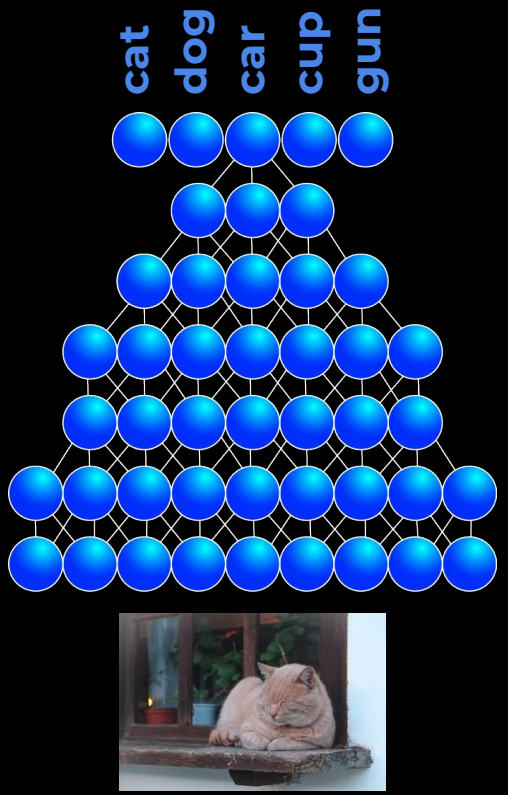
Deployment phase:

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

Excursion: Deep Learning

Deployment phase:

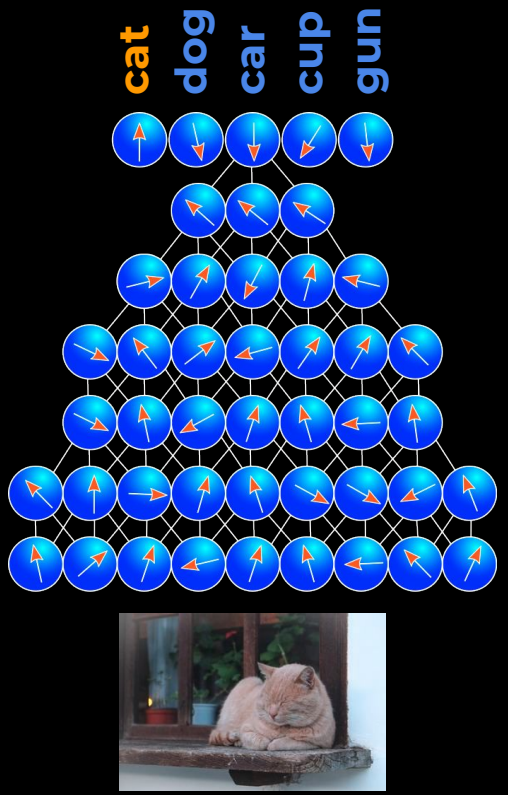
- connection weights remain fixed
- network can make predictions
- no more learning



Excursion: Deep Learning

Deployment phase:

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

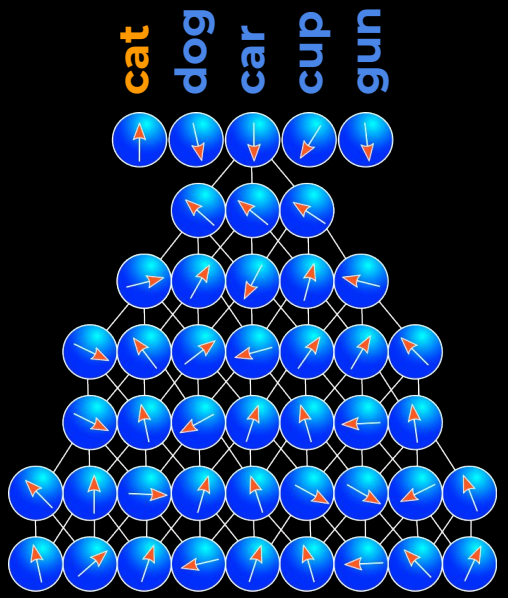


Excursion: Deep Learning

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.



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.

Robustness to unexpected or even adversarial conditions

Statistical learning theory: for any function f:

$$\underbrace{\mathbb{E}[\ell_f(x)]}_{\text{expected error on future data}} \leq \underbrace{\frac{1}{n}\sum_{i=1}^{n}\ell_f(x_i)}_{\text{observed error on training data}} + \underbrace{\frac{\text{const.}}{\sqrt{n}}}_{\text{sampling uncertainty}}$$

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.

Example: domain shift

The training data rarely perfectly reflects future data:

- data collection bias

- annotation bias

truck?
or
lorry?

- static-world bias

Example: adversarial examples

Image classification: recognize 1000 object categories in natural images

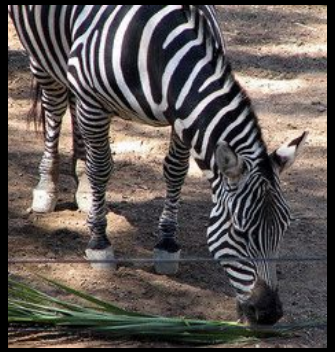
- state-of-the-art deep (convolutional) neural network
- trained on 1.2 millions images
- tested on 100.000 additional images:
  - predicts correct category on more than 80%
  - correct category in top-5 predictions in more than 95%

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:

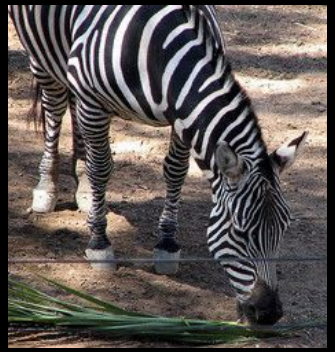


Example: adversarial examples

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

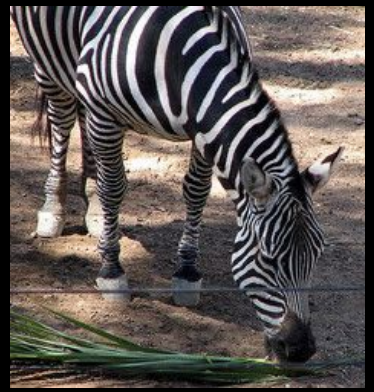
- state-of-the-art deep (convolutional) neural network
- trained on 1.2 millions images
- tested on 100.000 additional images:
  - predicts correct category on more than 80%
  - correct category in top-5 predictions in more than 95%

Example 2:

→ **toaster** (97% confidence)

Example: adversarial examples

Example 1

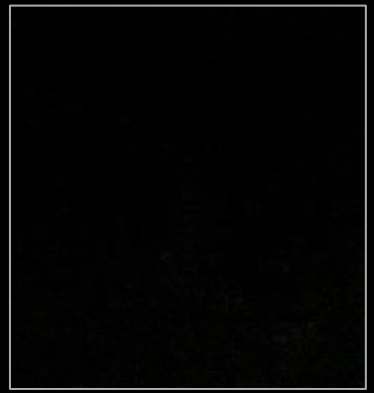


zebra

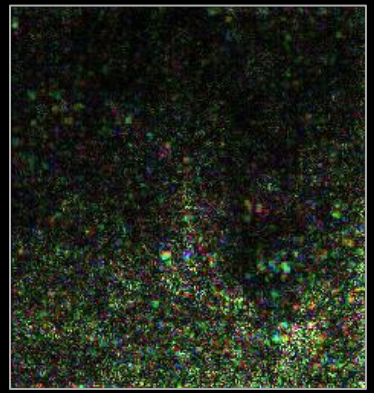
Example 2

toaster

Difference



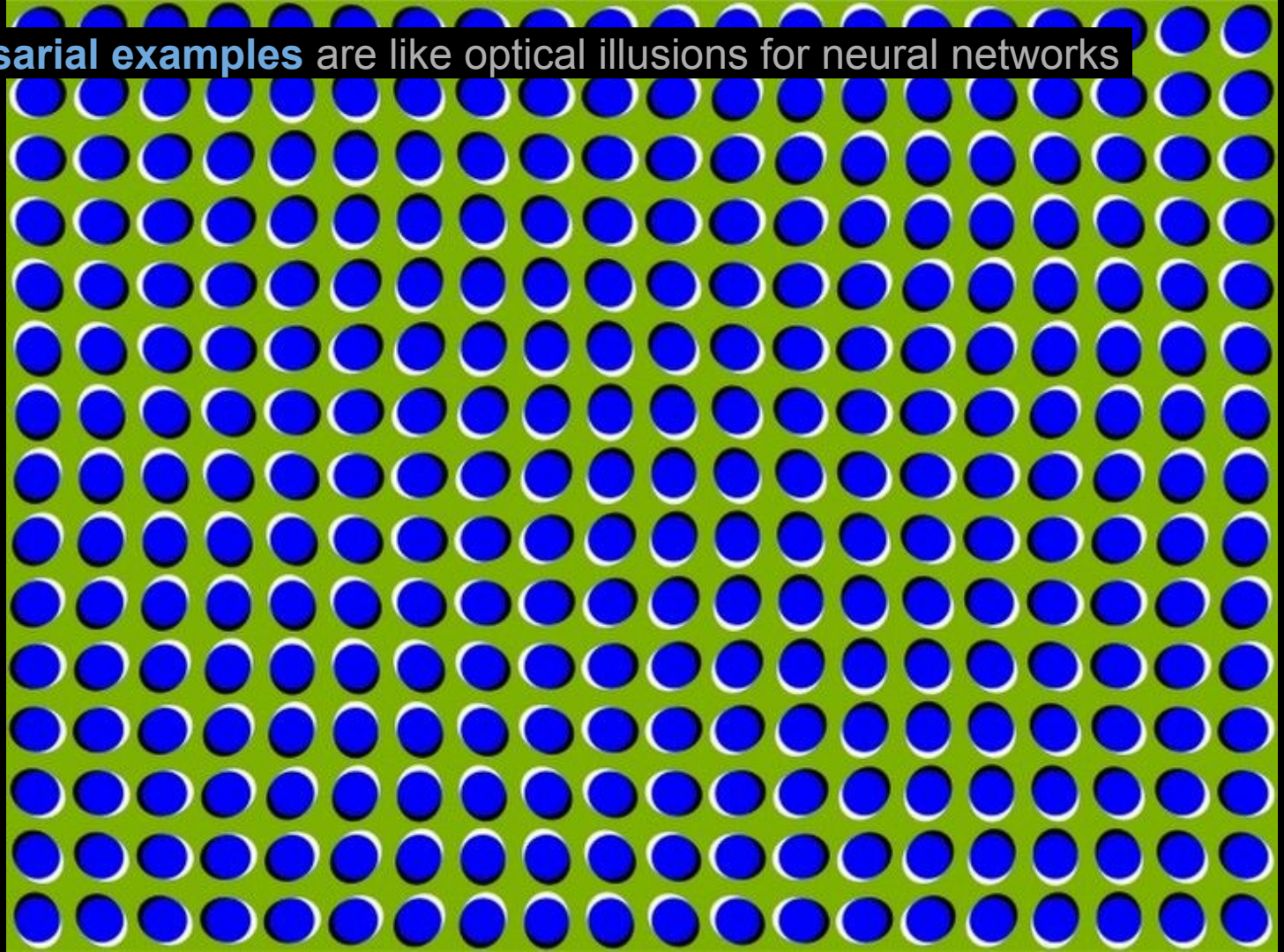
Difference (20x)



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

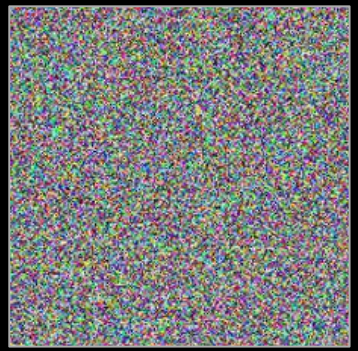


Overview

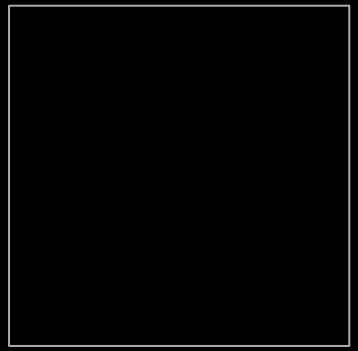
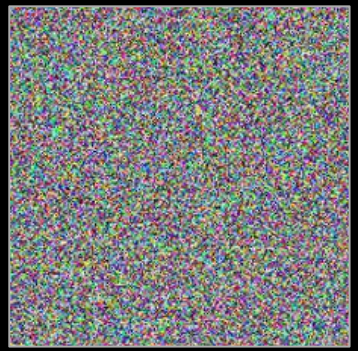
To become trustworthy, learning system should behave more human-like:

- **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

input image				
ideal output	`ski`	`ski` (rotated camera)	`nothing` (just black)	`nothing` (random noise)

Introspection, to be aware of its own performance, including failures

input image				
ideal output	`ski`	`ski` (rotated camera)	`nothing` (just black)	`nothing` (random noise)
prediction	`ski`	`paintbrush`	`web site`	`tennis ball`

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.”*

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.

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 !

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:

→ `hard disk`

After the training phase is over, networks are unable to learn from their mistakes.

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.

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.”*

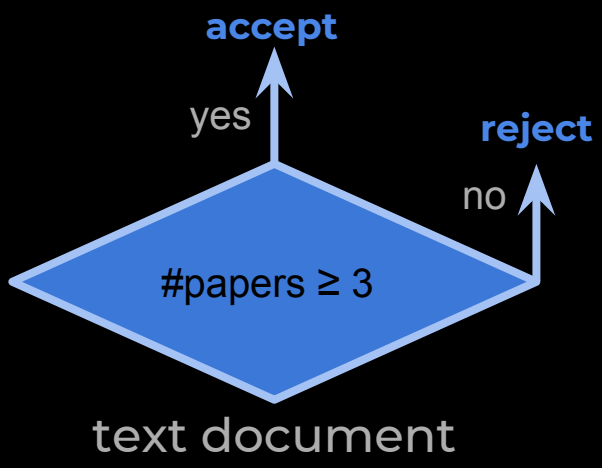
People don’t just want decisions, they want **explanations**.

Transparency, **explainability** and **fairness** of the decision process

Simple models

e.g. naive Bayes, decision trees, ...

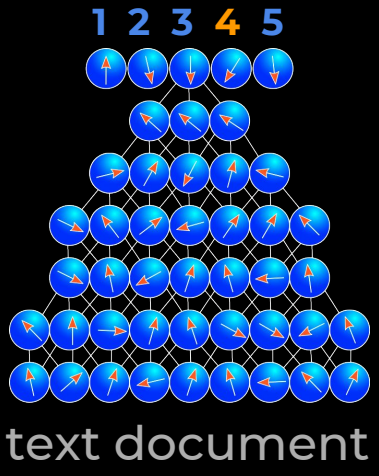
- are (often) easy to explain
- provide only limited accuracy



Complex models,

e.g. deep neural networks

- offer high classification accuracy
- decisions are hard/impossible to explain



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.”*

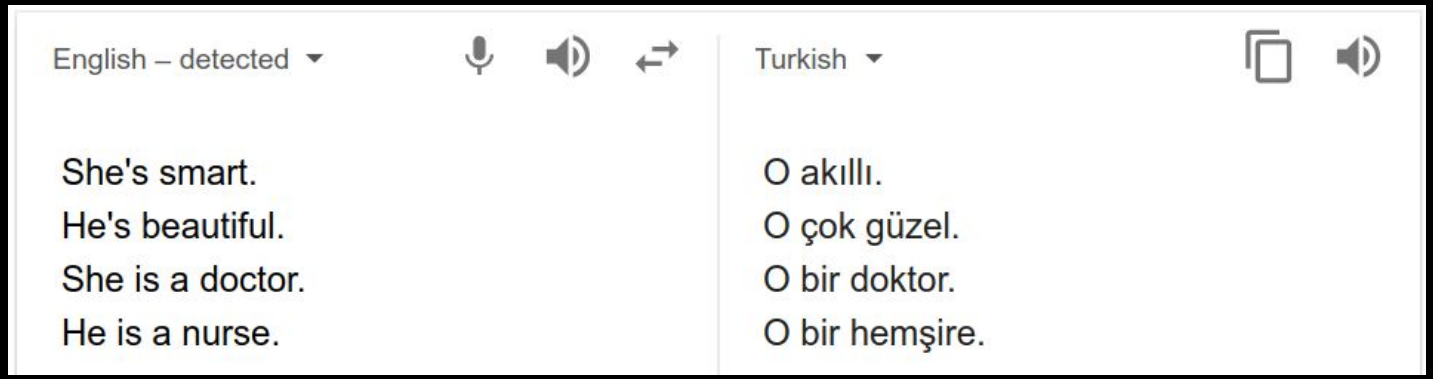
Scenario 4:

- 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.**

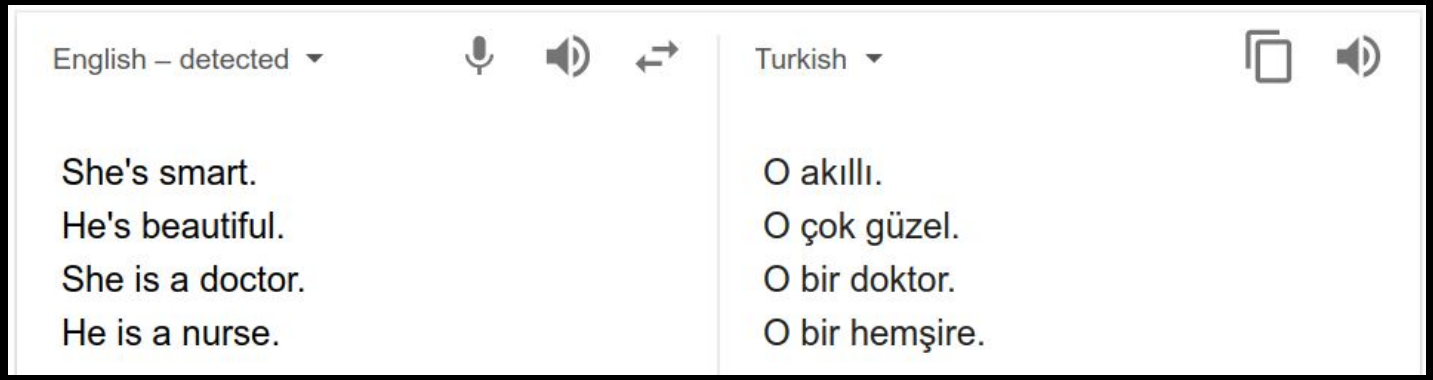
Transparency, **explainability** and **fairness** of the decision process

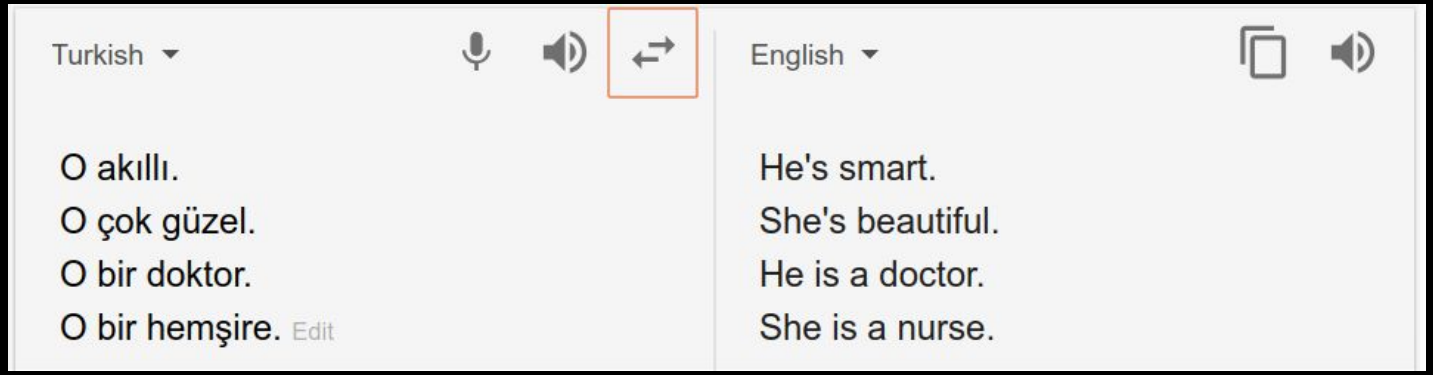
Example: Google Translate has a **gender bias**



Transparency, **explainability** and **fairness** of the decision process

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





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]

Remy Sun
(ENS Rennes)

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

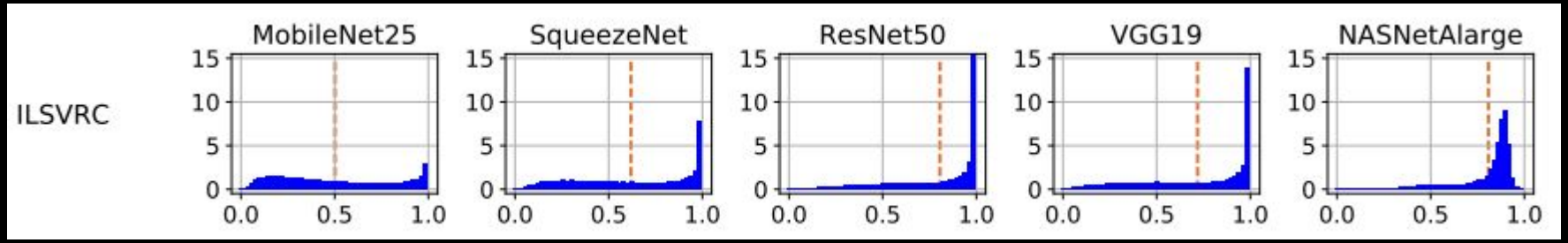
- work on batches (groups of images) instead of single images
  → allows us to control the false positive rate

- use Kolmogorov-Smirnov (KS) test to compare score distributions
  → efficient, distribution-free, well-understood null distribution

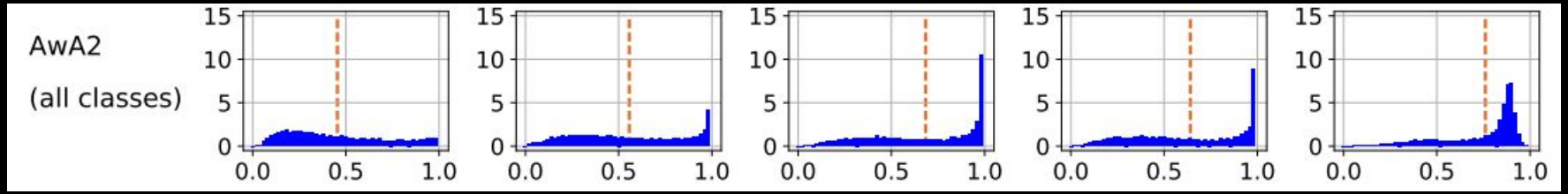
Introspection, to be aware of its own performance, including failures

Illustration: five different 1000-class object detection networks

Score distribution on original data:



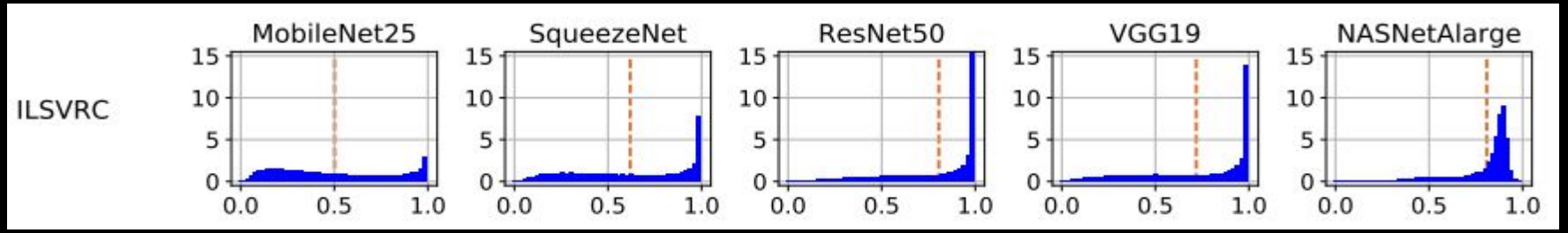
Score distribution on images of other classes (AwA2 dataset):



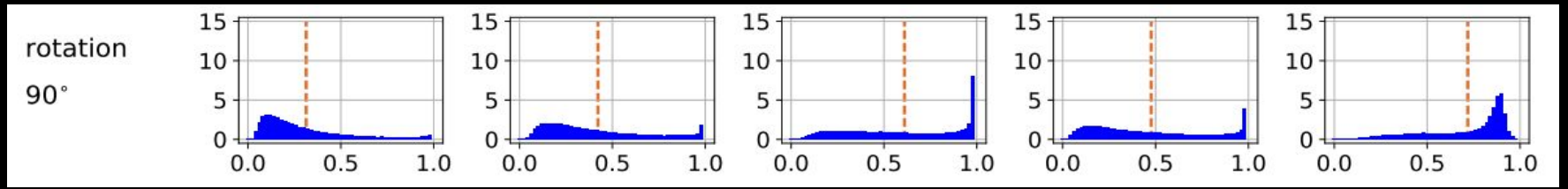
→ difference can be detected reliably

Introspection, to be aware of its own performance, including failures

Score distribution on original data:



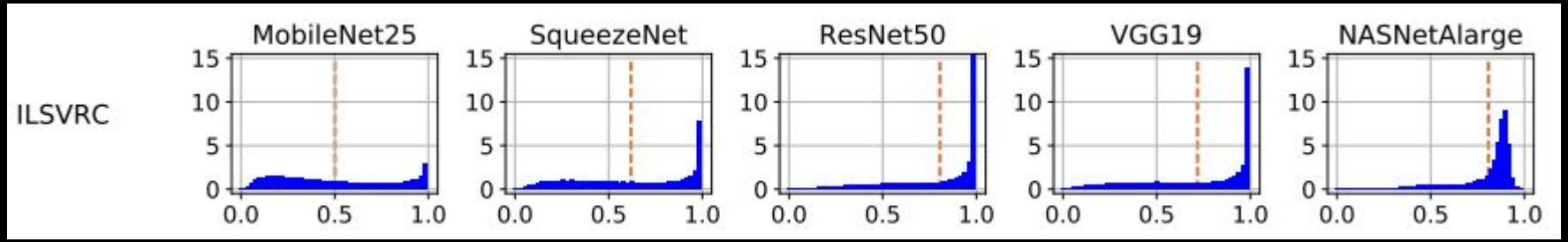
Score distribution on rotated images:



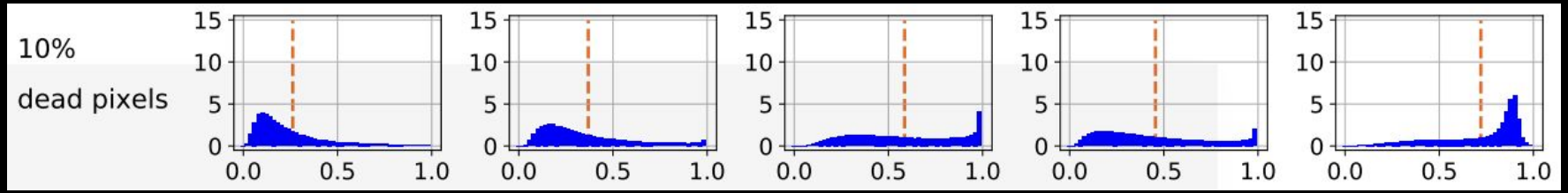
→ difference can be detected reliably

Introspection, to be aware of its own performance, including failures

Score distribution on original data:



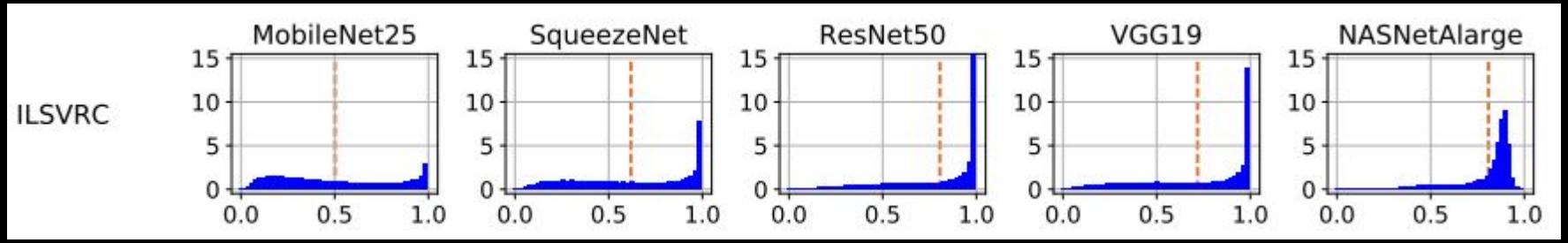
Score distribution on images with dead pixels:



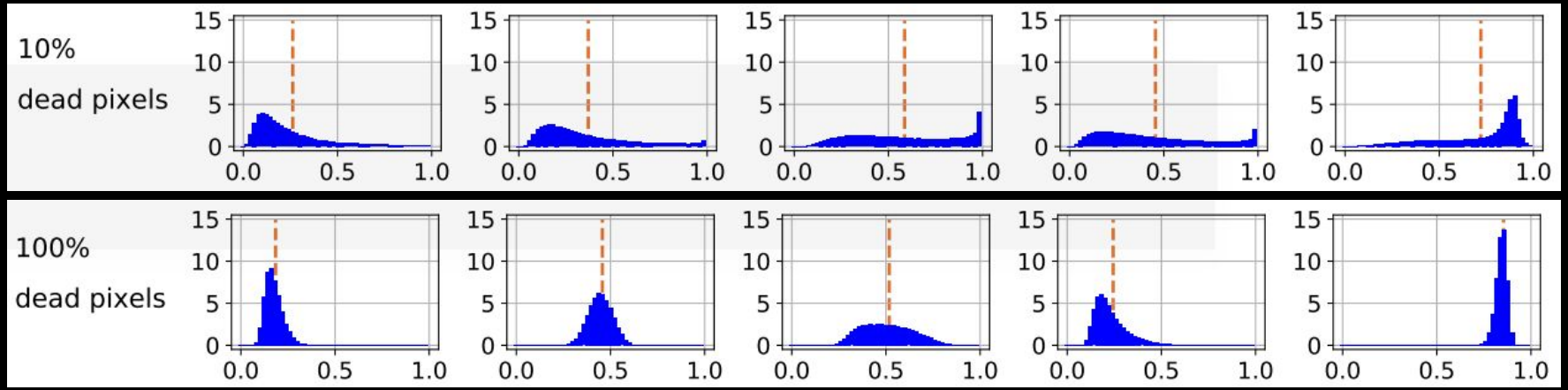
→ difference can be detected reliably

Introspection, to be aware of its own performance, including failures

Score distribution on original data:



Score distribution on images with dead pixels:



Introspection, to be aware of its own performance, including failures

Quantitative results (avg. detection rate):

	KS(conf)	mean	logmean	z	log-z	sym.mean	sym.logmean	sym.z	sym.log-z
AwA2-bat	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
AwA2-blue whale	1.00	0.20	0.00	0.20	0.00	0.80	0.00	0.80	0.62
AwA2-bobcat	1.00	0.00	0.00	0.00	0.00	1.00	0.00	1.00	1.00
AwA2-dolphin	1.00	1.00	0.79	1.00	0.79	1.00	0.66	1.00	0.70
AwA2-giraffe	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
AwA2-horse	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
AwA2-rat	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
AwA2-seal	1.00	0.20	0.20	0.20	0.20	1.00	0.20	1.00	1.00
AwA2-sheep	1.00	0.01	0.00	0.00	0.00	0.65	0.00	0.63	0.82
AwA2-walrus	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
DAVIS	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

Adaptivity to new situations or goals

iCaRL: Incremental Classifier and Representation Learning

Sylvestre Rebuffi
(U Oxford)

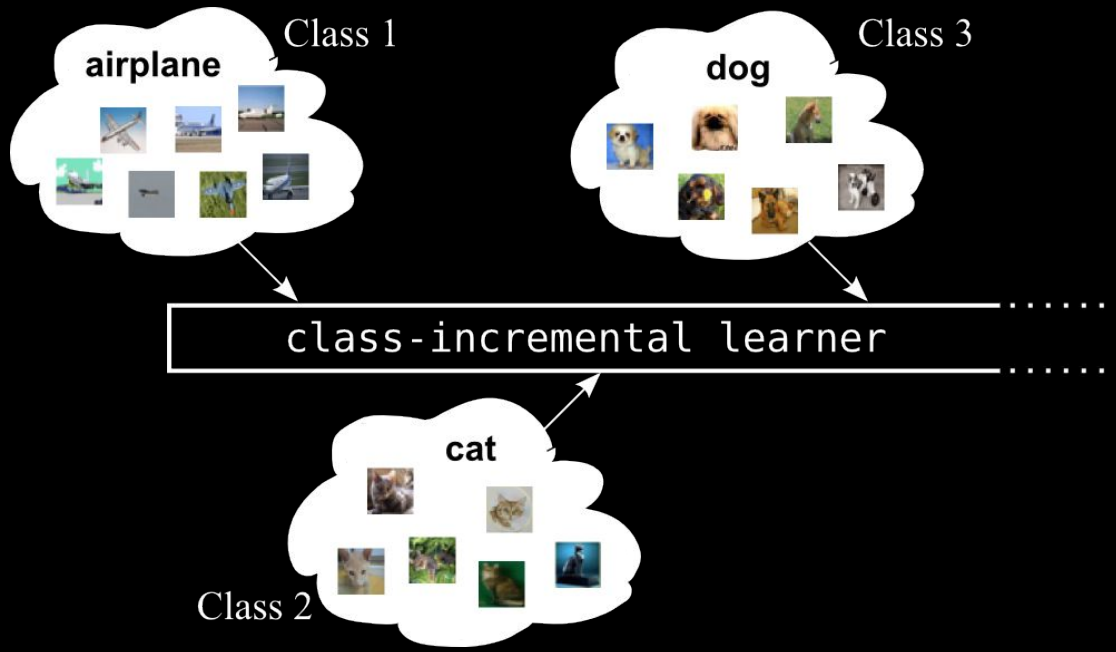
Alex Kolesnikov
(IST Austria)

Georg Sperl
(IST Austria)

CVPR 2017

arXiv:1611.07725 [cs.CV]

Adaptivity to new situations or goals



Images: CIFAR dataset

Setup: training data arrives one class at a time
the system can only store a (small) fixed-size amount of it

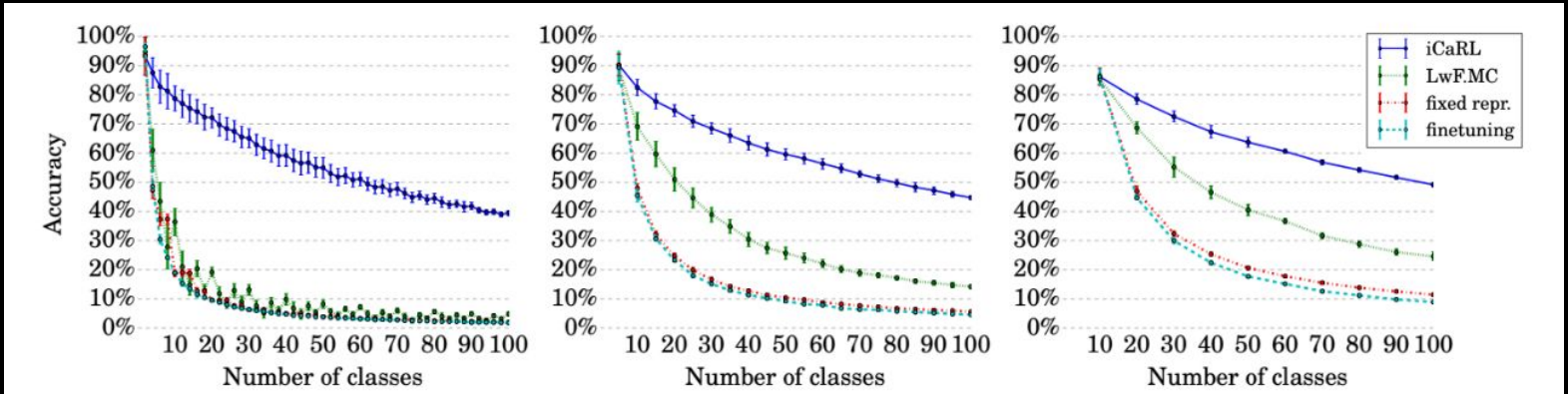
Goal: learn multi-class classifier, but avoid catastrophic forgetting

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:



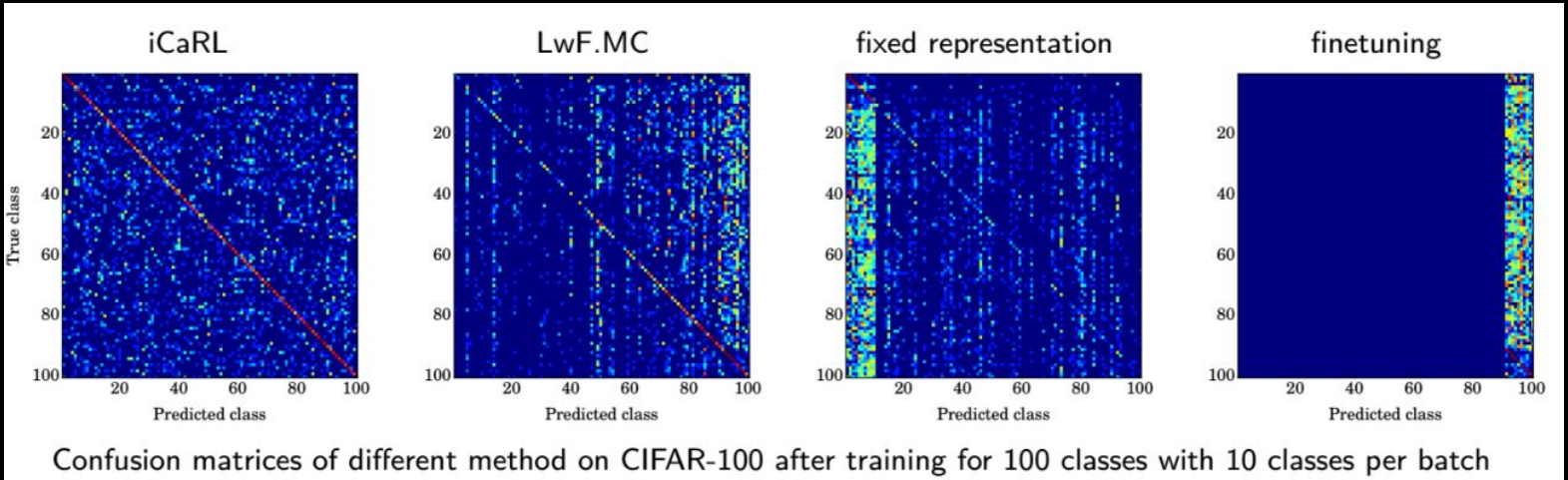
Multi-class accuracies over 10 repeats (average and standard deviation) for class-incremental training on CIFAR-100

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      ?
```
→ should be positive

```
The crib had sharp edges.
 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 $\beta \in [0, \alpha/4]$*

$$\mathbb{P}\left[R_n(h_n) - \inf_{h \in \mathcal{H}} R_n(h) > \alpha + 2\varepsilon\right] \tag{10}$$
$$\leq \frac{2k\mathcal{N}_\infty(\mathcal{L}(\mathcal{H}), \beta, n)}{(\alpha - 4\beta)^2} e^{-\frac{1}{2}m(\alpha-4\beta)^2} + \mathbb{P}\left[E_{k,m}^c\right].$$

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

$$\mathbb{P}\left[R_n(h_n) - \inf_{h \in \mathcal{H}} R_n(h) > \alpha + W_{I_n,n}/s_{I_n,n} + 4\varepsilon\right] \tag{13}$$
$$\leq \frac{4k\mathcal{N}_\infty(\mathcal{L}(\mathcal{H}), \beta, n)}{(\alpha/2 - 4\beta)^2} e^{-\frac{1}{2}m(\alpha/2-4\beta)^2} + \mathbb{P}\left[E_{k,m}^c\right]. \tag{14}$$

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 ε-close subroutine or a newly started one
  – identify all ε-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

$$h_a \leftarrow \mathsf{output}(S_j) \quad \text{for} \quad j = \operatorname*{argmin}_{j \in T} M_{t,j}$$

- **output** the currently active hypothesis, $h_t \leftarrow h_a$
- **observe** the next value of the process, $\mathbf{z}_t$
- **update** all ε-close subroutines:

$$S_j \leftarrow \mathsf{update}(S_j, \mathbf{z}_t) \quad \text{for all} \quad j \in J$$

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...

Thanks to...

Funding sources:

erc

IST AUSTRIA
Institute of Science and Technology

Group members and alumni:

Sylvestre Rebuffi

Asya Pentina

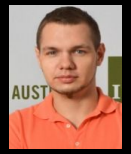
Viktoriiia Sharmanska

Alex Kolesnikov

Remy Sun

Amelie Royer

Nikola Konstantinov


Alex Zimin

Mary Phuong

Georg Sperl