Imperial College London

Imperial College London

Machine Learning for Imaging
Trustworthy AI/ML

Daniel Rueckert
Department of Computing
Imperial College London, UK

Trustworthy AI/ML

Seven key requirements for trustworthy AI/ML

https://ec.europa.eu/digital-single-market/en/news/ethics-guidelines-trustworthy-ai

The need for data

- The power and effectiveness of ML is critically dependent on the data that is used to train ML models (particularly for deep learning models)
  - Quality of the data is one of the important aspects that determines the effectiveness of deep learning models:
    - Curation of the data (and the associated annotations)
    - Representativeness of the data
  - Quantity of data

- In general, the more data is available for training, the more accurate and robust the resulting ML models become.
  - Data sharing is more important, not only for training ML models but also for evaluating ML solutions in multi-institutional/multi-national trials

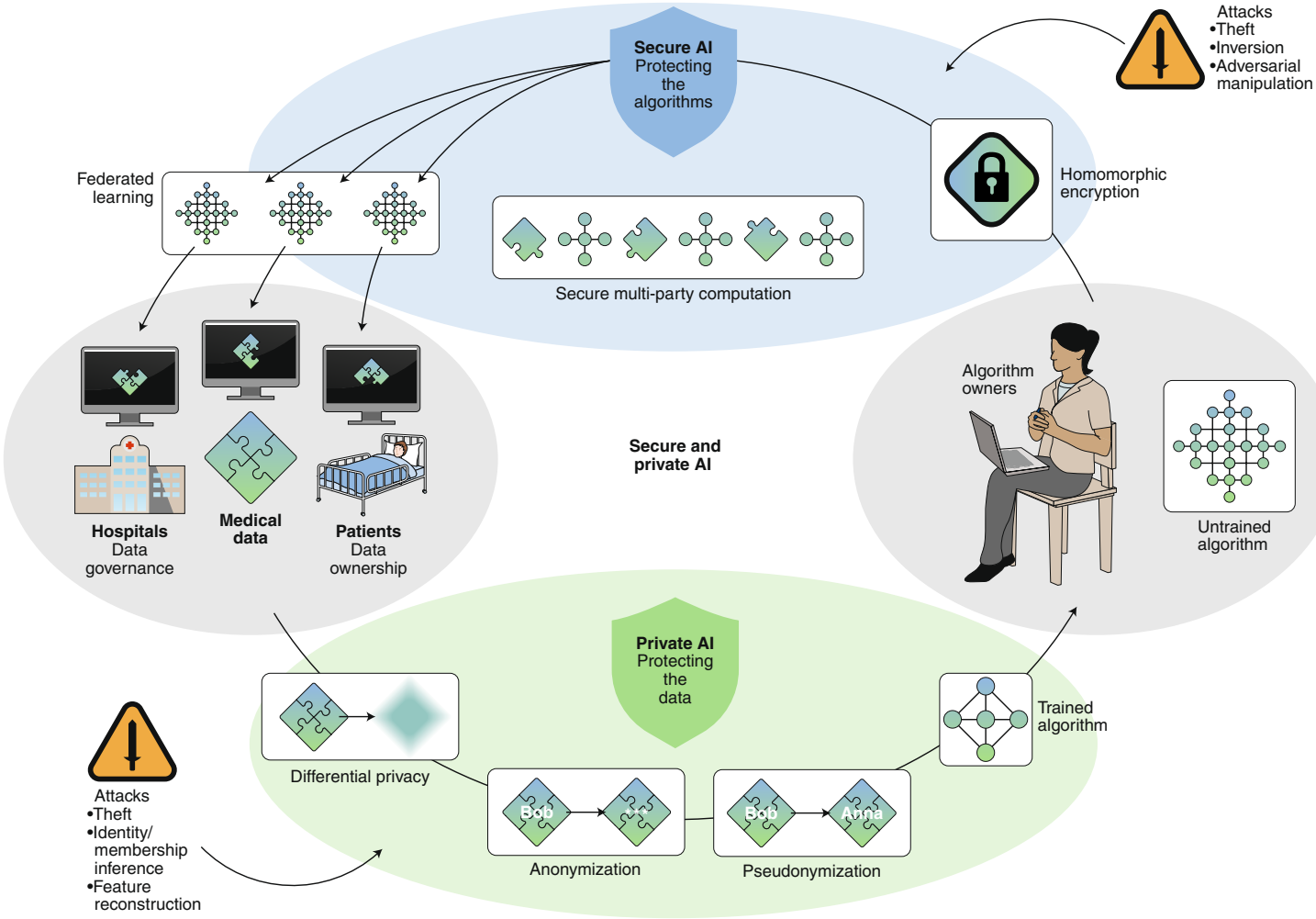
Deep learning: What are the hurdles to getting more data?

- Human and societal challenges
  - Cost and effort for collecting and annotating data
  - Incentives for data sharing (money, fame, other benefits)
- Technical challenges
  - Data quality
  - Data annotation
  - Data exchange formats
- Legal challenges
  - What is allowed? What consent is required?
  - Regulation (e.g. GDPR)
- Privacy challenges
  - Ethical
  - Trust (risks such as privacy breaches, data leaks and re-identification)

Key principles of GDPR

- Lawfulness, fairness and transparency
- Purpose limitation
- Data minimisation
- Accuracy
- Storage limitation
- Integrity and confidentiality
- Accountability

Secure and Privacy-aware ML



Secure and privacy-preserving ML

- Optimal privacy preservation requires implementations that are secure by default so-called *privacy by design*

- Requirements:
  - Minimal or no data transfer
  - Provision of theoretical and/or technical guarantees of privacy

Federated learning: train a ML model across decentralized clients with local data, without exchanging them

Differential privacy: perturb the data so that information about the single individual is reduced while retaining the capability of learning

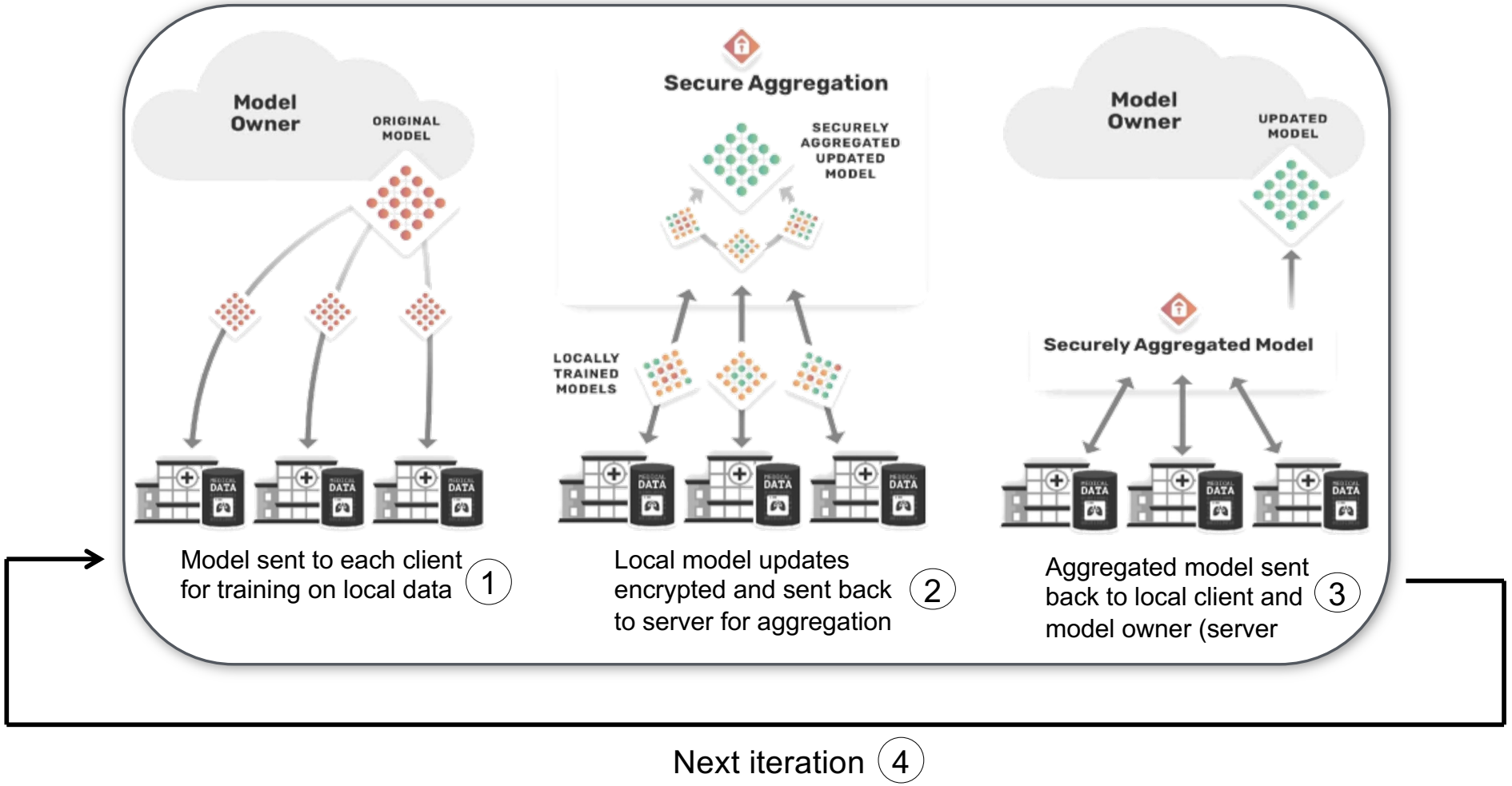
Secure and privacy-preserving ML

- Optimal privacy preservation requires implementations that are secure by default so-called *privacy by design*

- Requirements:
  - Minimal or no data transfer
  - Provision of theoretical and/or technical guarantees of privacy

- Other approaches for privacy-preserving AI:
  - Homomorphic encryption which enables learning from encrypted data
  - Secure multi-party computing where processing is performed on encrypted data shares, split among them in a way that no single party can retrieve the entire data on their own.
  - Trusted execution environments

Federated learning



Federated learning – in detail

Recap:

- For a training dataset containing n samples (x_i, y_i) with $1 \leq i \leq n$, the loss function can be written as:

$$\mathcal{L} = \frac{1}{n} \sum_i \mathcal{L}_i(\boldsymbol{x}_i, y_i, \theta)$$

Loss for one example i given current model parameters θ

- Optimization uses gradient descent, e.g.

$$\Theta^{j+1} := \Theta^j - \tau \, \boldsymbol{\nabla} \, \mathcal{L}(\Theta)$$

- Typical gradient descent uses SGD and its variants, through mini-batches of size m

$$\boldsymbol{\nabla} \, \mathcal{L}(\Theta) \approx \sum_{i=1}^{m} \boldsymbol{\nabla} \mathcal{L}(\boldsymbol{x}_i, y_i, \Theta)$$

Federated learning – in detail

In federated learning:

- Suppose N training samples are distributed to K clients, P_k is the set of indices of samples at client k, and $n_k = |P_k|$

$$\mathcal{L}(\Theta) = \sum_{k=1}^{K} \frac{n_k}{N} \mathcal{L}_k(\Theta)$$

with

$$\mathcal{L}_k(\Theta) = \frac{1}{n_k} \sum_{i \in P_k} \mathcal{L}(\boldsymbol{x}_i, y_i, \Theta)$$

$\mathbb{E}_{P_k}[\mathcal{L}_k] = \mathcal{L}$ iid setting

$\mathbb{E}_{P_k}[\mathcal{L}_k] \neq \mathcal{L}$ non-iid setting

Federated SGD

In federated learning:

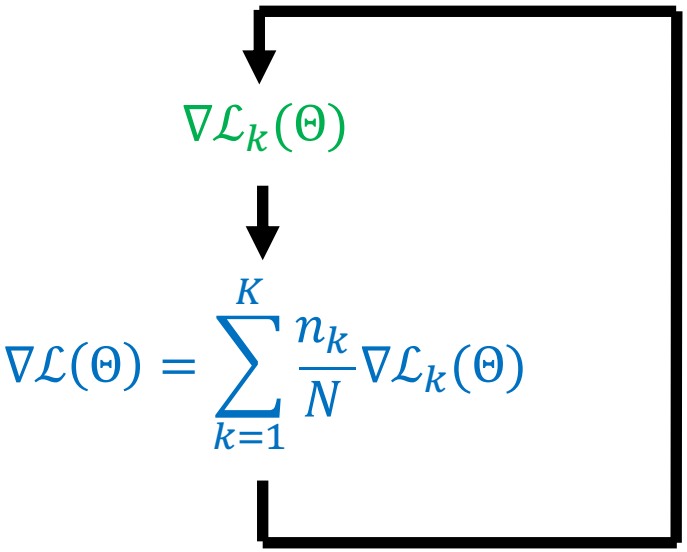
- Suppose a C fraction of clients are selected at each round:

  $C = 1$: full-batch (non-stochastic) gradient descent

  $C < 1$: stochastic gradient descent (SGD)

- Each client computes:

- Server computes:



Federated SGD

In federated learning:

- Suppose a C fraction of clients are selected at each round:

  $C = 1$: full-batch (non-stochastic) gradient descent

  $C < 1$: stochastic gradient descent (SGD)

- Each client computes:

$$\nabla \mathcal{L}_k(\Theta)$$

- Server computes:

$$\Theta_k^{j+1} = \Theta_k^j + \tau \sum_{k=1}^{K} \frac{n_k}{N} \nabla \mathcal{L}_k(\Theta)$$

Federated Averaging

In federated learning:

- Suppose a C fraction of clients are selected at each round:

  $C = 1$: full-batch (non-stochastic) gradient descent

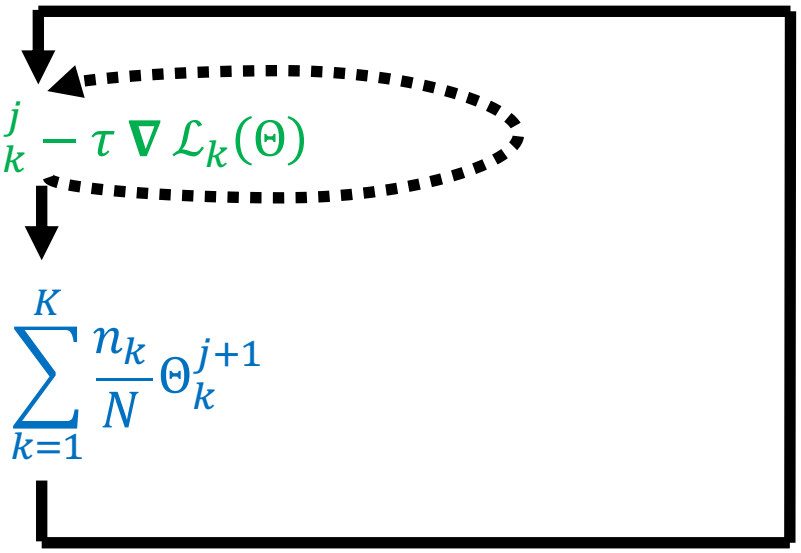
  $C < 1$: stochastic gradient descent (SGD)

- Each client computes:

$$\Theta_k^{j+1} := \Theta_k^j - \tau \nabla \mathcal{L}_k(\Theta)$$

- Server computes:

$$\Theta_k^{j+1} = \sum_{k=1}^{K} \frac{n_k}{N} \Theta_k^{j+1}$$



Federated Averaging

Algorithm 1 `FederatedAveraging`. The K clients are indexed by k; B is the local minibatch size, E is the number of local epochs, and η is the learning rate.

Server executes:
initialize w_0
for each round $t = 1, 2, \dots$ **do**
  $m \leftarrow \max(C \cdot K, 1)$
  $S_t \leftarrow$ (random set of m clients)
  for each client $k \in S_t$ **in parallel do**
    $w_{t+1}^k \leftarrow \text{ClientUpdate}(k, w_t)$
  $w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k$

ClientUpdate(k, w): *// Run on client k*
$\mathcal{B} \leftarrow$ (split $\mathcal{P}_k$ into batches of size B)
for each local epoch i from 1 to E **do**
  for batch $b \in \mathcal{B}$ **do**
    $w \leftarrow w - \eta \nabla \ell(w; b)$
return w to server

- First, model is randomly initialized on the central server
- For each round t:
  - *A random set of clients are chosen;*
  - *Each client performs local gradient descent steps;*
  - *The server aggregates model parameters submitted by the clients.*

Challenges for federated learning

- Non-IID data
  - Training data for a given client is typically site specific, hence the site's local dataset will not be representative of the distribution of training samples.
- Unbalanced data
  - Sites may have a lot or little training data, leading to varying amounts of local training data across different sites.
- Massively distributed data
  - There may be extreme scenarios where each site only has very training samples (in the limiting case one example)
- Communication costs
  - Communication between clients and servers occurs communication overheads. The amount of overhead will depend on the number of clients and the frequency of updates from/to server.

Homomorphic Encryption

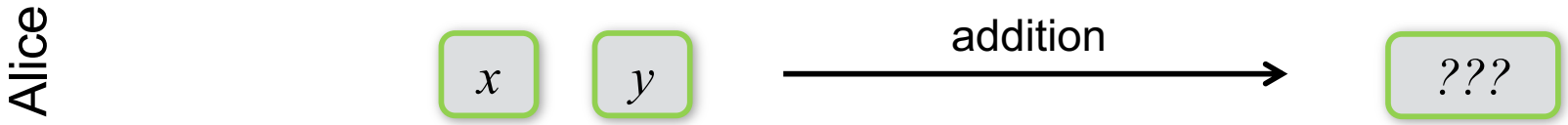
- Based on the assumption that one can perform (basic) arithmetic on encrypted values, i.e.

$$[x] \oplus [y] = [x + y] \qquad \text{and} \qquad [x] \otimes [y] = [x \cdot y]$$

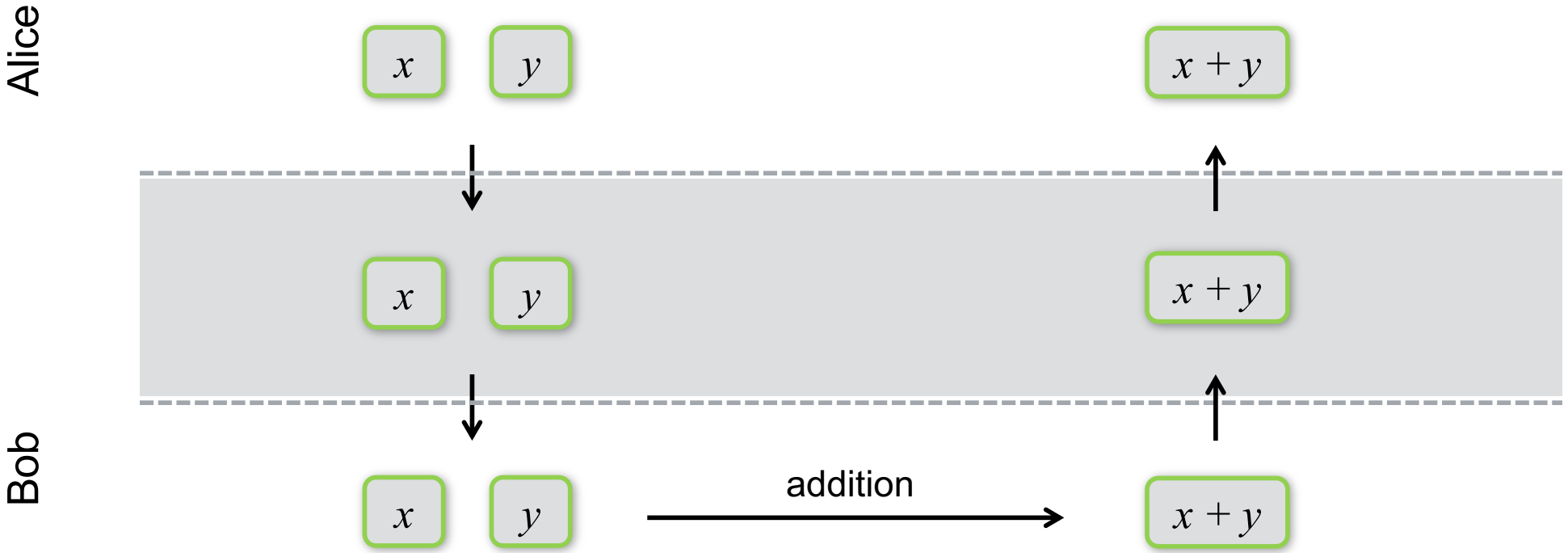
- Here:

$[xyz]$	encryption of some plaintext xyz
$\oplus$	homomorphic addition operation in ciphertext space
$\otimes$	homomorphic multiplication operation in ciphertext space

ML in standard setting

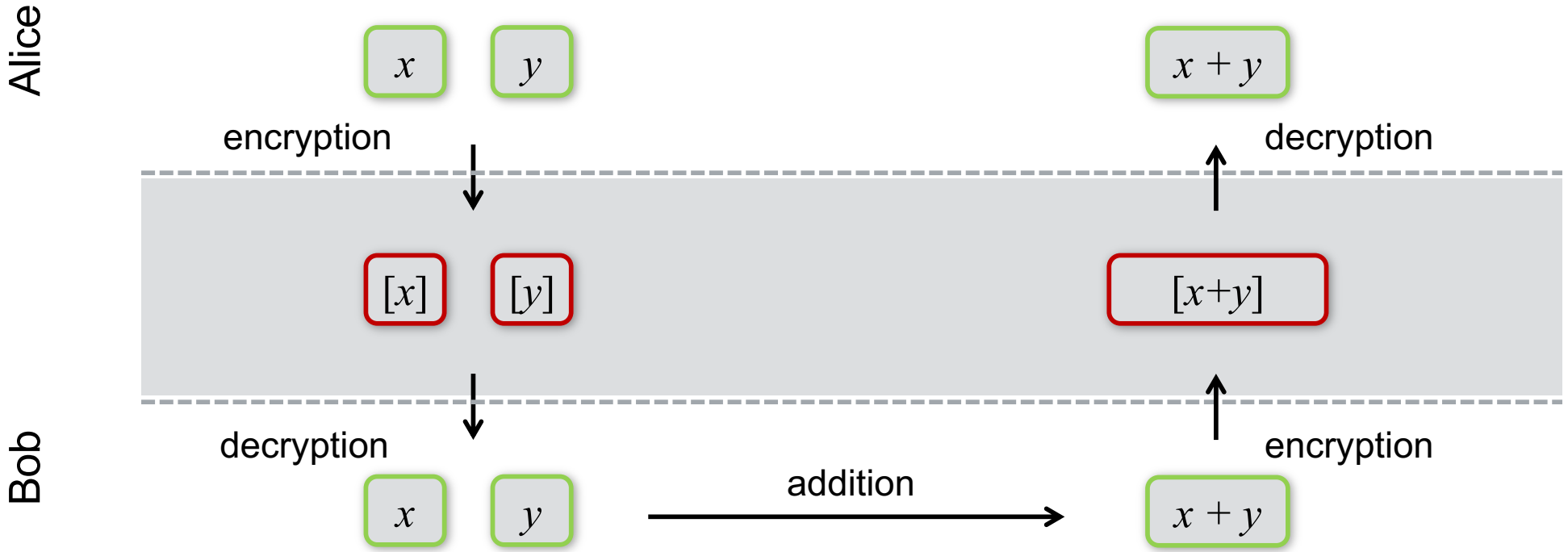


ML in client/server setting



Data is transmitted in plain text

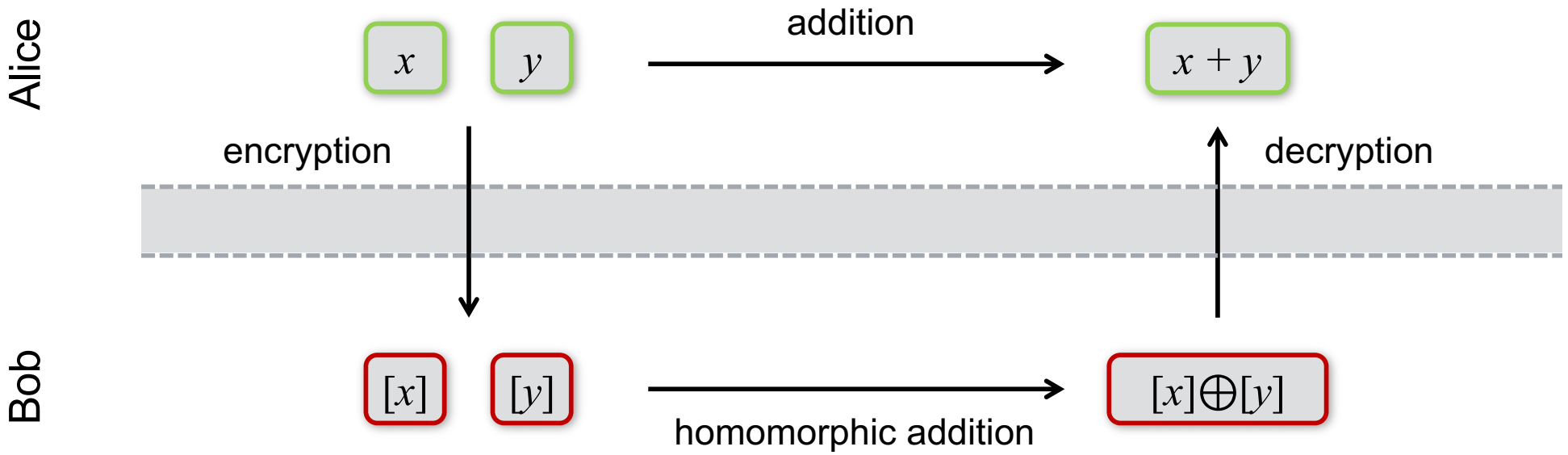
ML in client/server setting



Data is transmitted in encrypted form

Bob does see data in unencrypted form

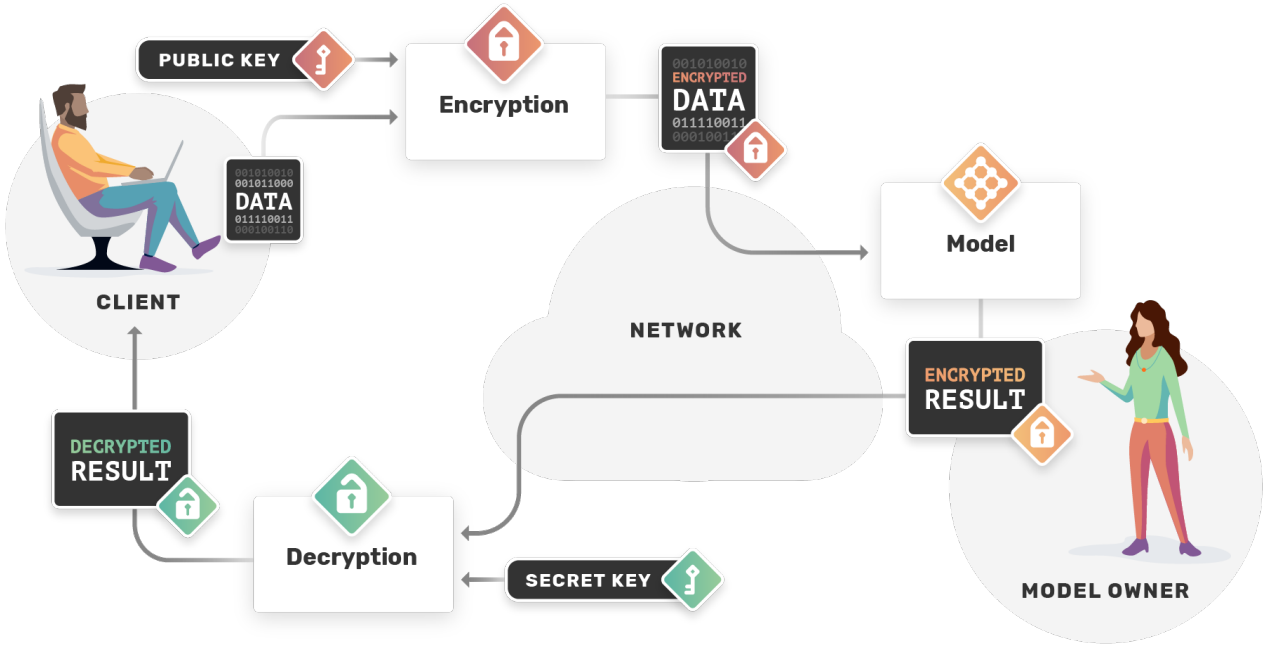
Homomorphic Encryption



Data is transmitted in encrypted form
Bob does not see data in unencrypted form

ML and homomorphic encryption

- Outsourced computation
- Private prediction
- Private training



ML and homomorphic encryption

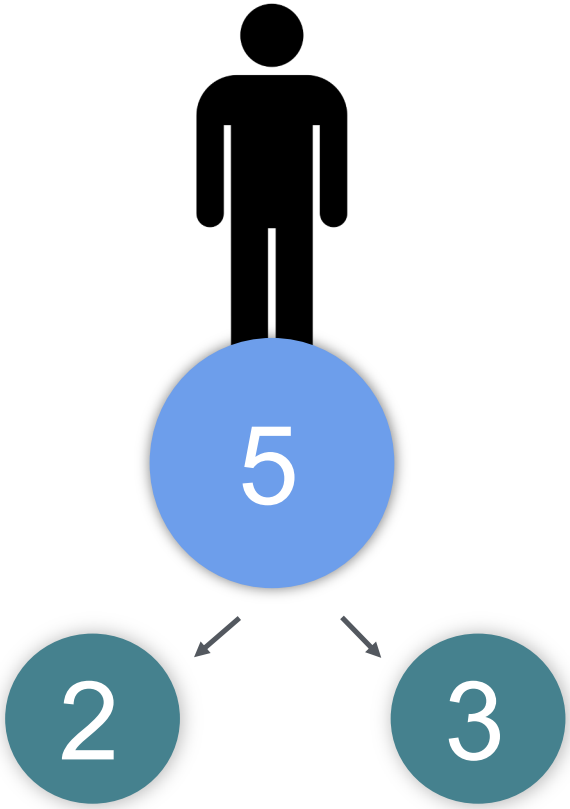
Advantages

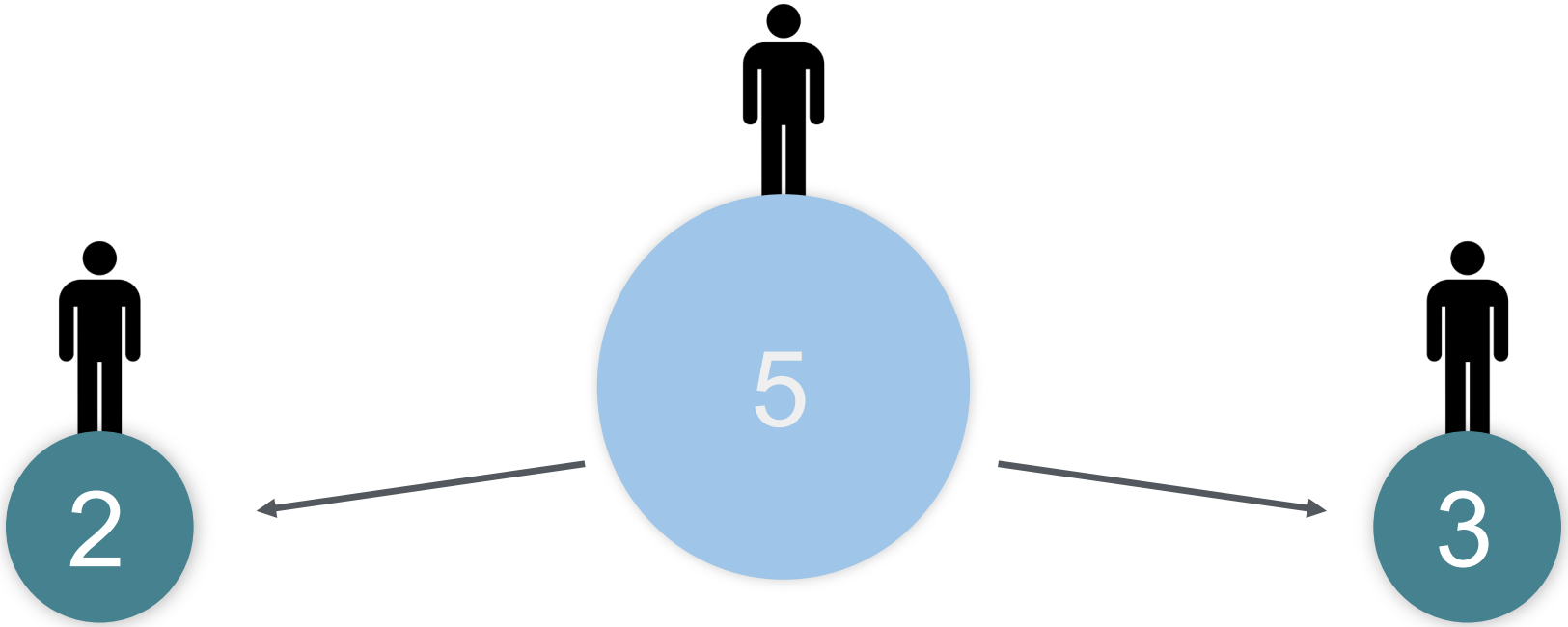
- **Can perform inference on encrypted data,** i.e. model owner never sees the client's private data
- **Does not require interaction** between data and model owners

Disadvantages

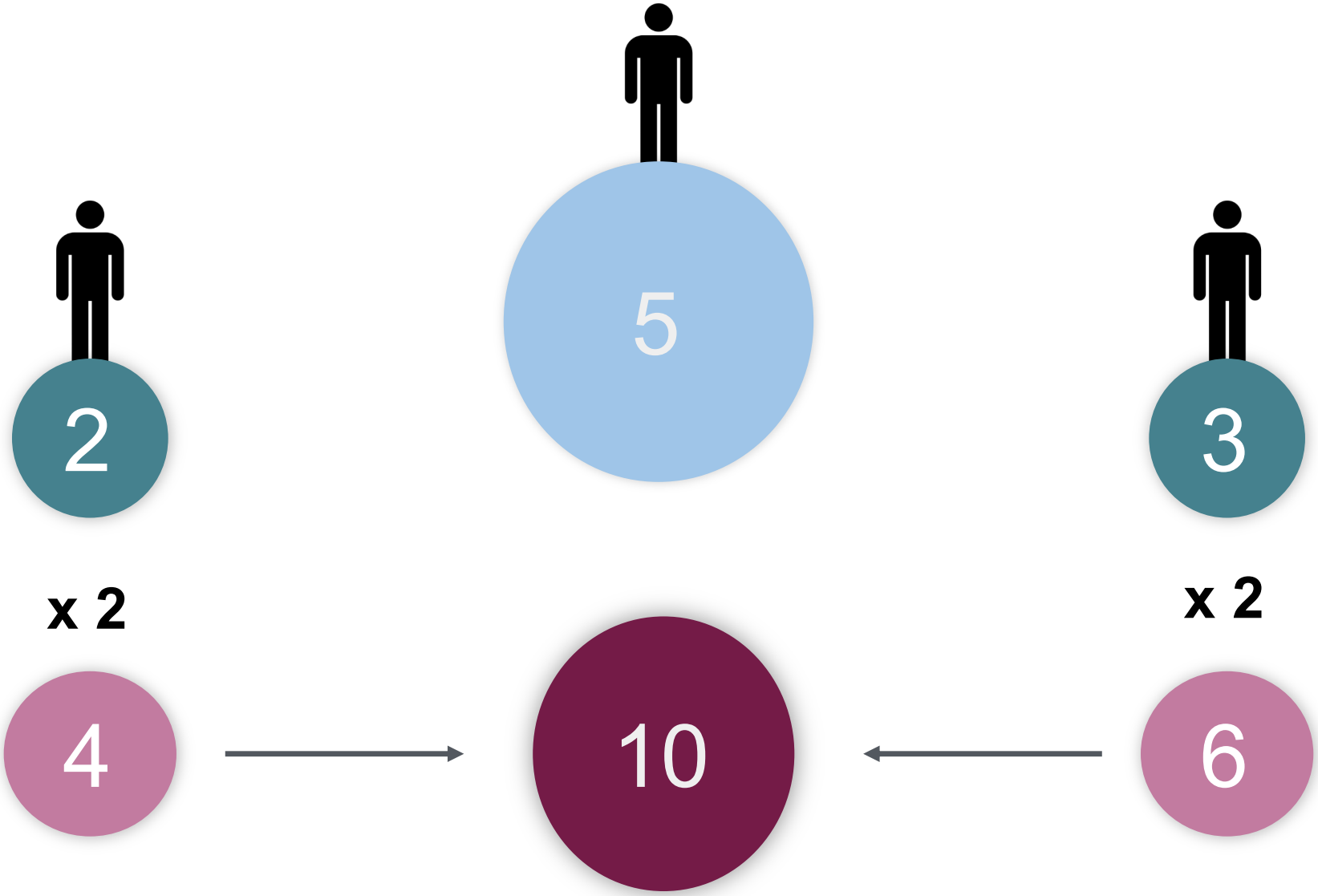
- **Computationally expensive** as it introduces a significant overhead
- **Restricted to certain operations** (also because of efficiency concerns)

Secure Multi-Party Computation



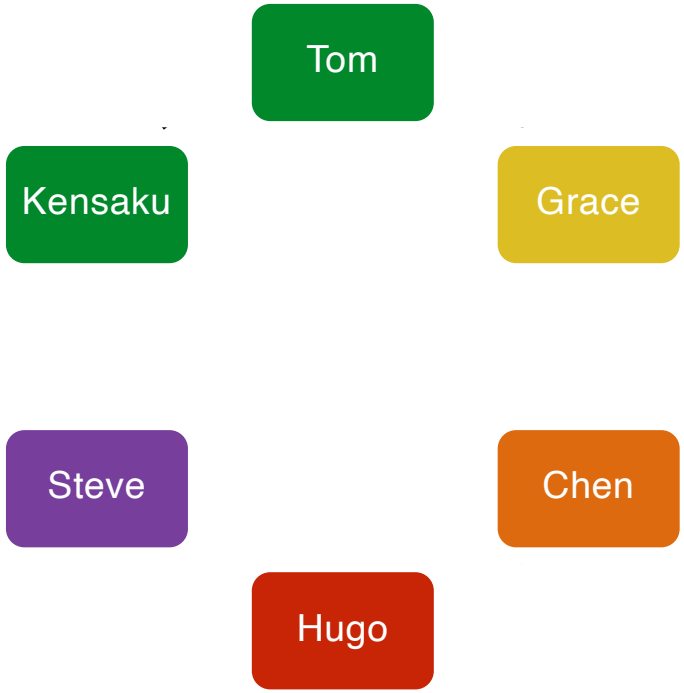


- **Confidentiality:** neither knows the real value
- **Shared Governance:** The value can only be disclosed if everyone agrees


5
2
3
x 2
x 2
4
10
6

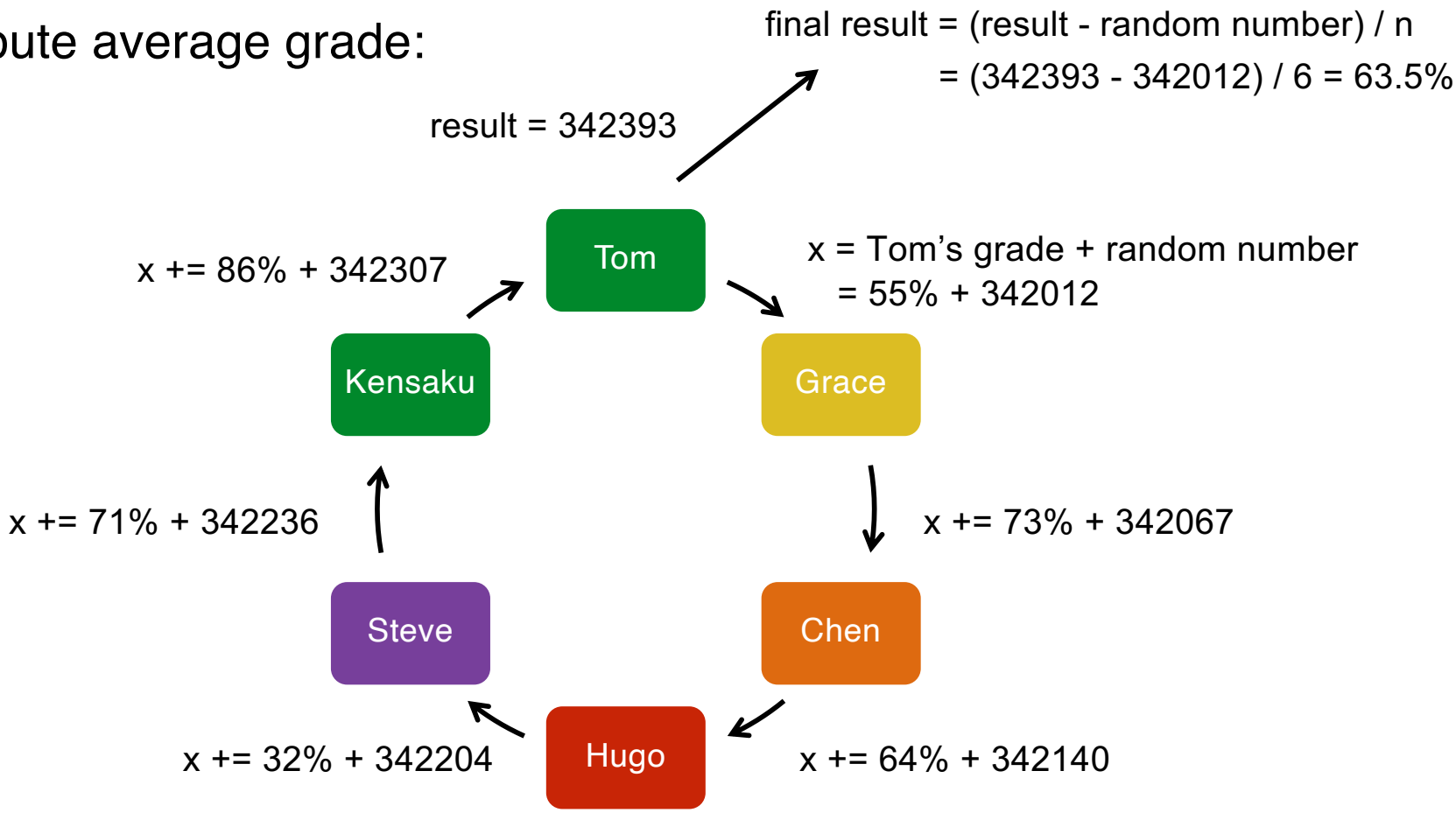
Secure Multi-Party Computation

- Compute average grade:



Secure Multi-Party Computation

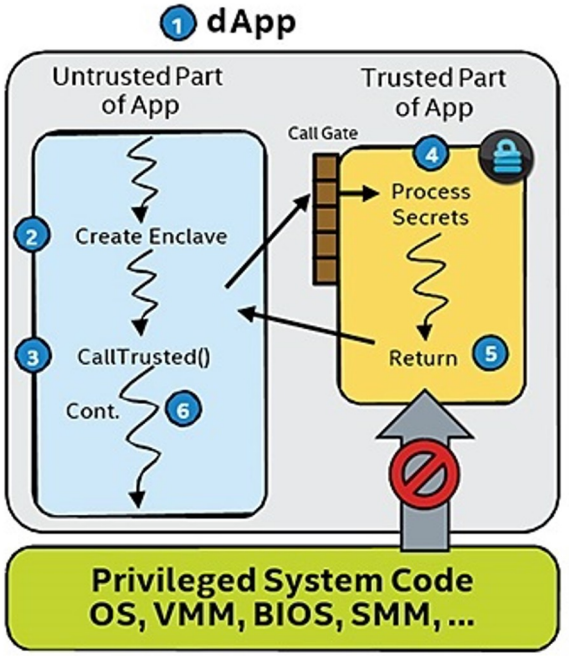
- Compute average grade:



Trusted Execution Environments

- Set of **CPU instructions** to create enclaves in RAM, that **no one can access** - except code from the enclave itself

- Ensures **total confidentiality of data during computation** - decryption happens only inside the enclave



Confidential Computing BETA

Encrypt data in-use with Confidential VMs. Available in Beta for Google Compute Engine.

What is privacy?

Anonmyzation

- The most straightforward approach is anonymization where identifying information is removed.
  - For example, a name may be removed from a medical record.

- Unfortunately, anonymization is rarely sufficient to protect privacy as the remaining information can be uniquely identifying.
  - For instance, given the gender, postal code, age, ethnicity and height, it may be possible to identify someone uniquely, even in a very large database.

k-anonymity

- One approach to prevent linkage attacks is k-anonymity.
  - A dataset is said to be k *anonymous* if, for any person's record in a dataset, there are at least k –1 other records which are indistinguishable.
  - So if a dataset is k-anonymous, then the best a linkage attack could ever do is identify a group of k records which could belong to the person of interest.
  - Even if a dataset isn't inherently k-anonymous, it could be made so by removing entire fields of data (like names and addresses) and selectively censoring fields of individual people who are particularly unique.

k-anonymity: Example

Original Database to Disclose

	IDENTIFYING VARIABLE	QUASI-IDENTIFIERS		
	ID	Name	Gender	Year of Birth	Test Result
1	1	John Smith	Male	1959	+ve
.	2	Alan Smith	Male	1962	-ve
.	3	Alice Brown	Female	1955	-ve
.	4	Hercules Green	Male	1959	-ve
.	5	Alicia Freds	Female	1942	-ve
.	6	Gill Stringer	Female	1975	-ve
.	7	Marie Kirkpatrick	Female	1966	+ve
.	8	Leslie Hall	Female	1987	-ve
.	9	Bill Nash	Male	1975	-ve
.	10	Albert Blackwell	Male	1978	-ve
.	11	Beverly McCulsky	Female	1964	-ve
.	12	Douglas Henry	Male	1959	+ve
.	13	Freda Shields	Female	1975	-ve
s|	14	Fred Thompson	Male	1967	-ve

2-Anonymization

	QUASI-IDENTIFIERS		
	ID	Gender	Decade of Birth	Test Result
1	1	Male	1950-1959	+ve
.	2	Male	1960-1969	-ve
.	4	Male	1950-1959	-ve
.	6	Female	1970-1979	-ve
.	7	Female	1960-1969	+ve
.	9	Male	1970-1979	-ve
.	10	Male	1970-1979	-ve
.	11	Female	1960-1969	-ve
.	12	Male	1950-1959	+ve
.	13	Female	1970-1979	-ve
|s'|	14	Male	1960-1969	-ve

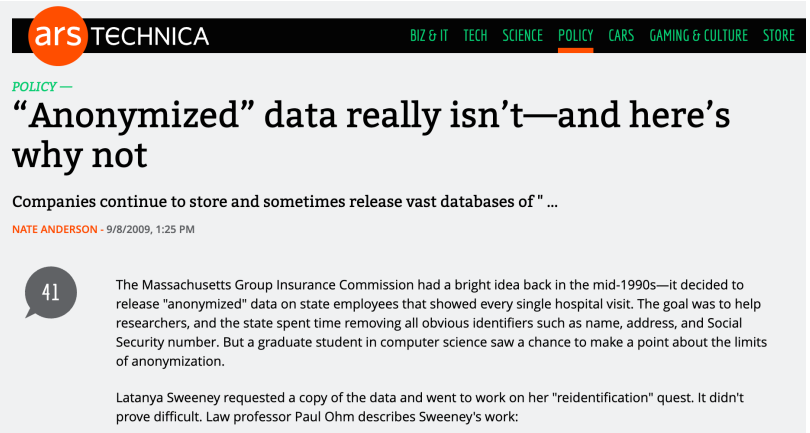
Disclosed (k-Anonymized) Database (ζ)

K. El Emam et al. Protecting privacy using k-anonymity. J Am Med Inform Assoc. 2008
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2528029/

k-anonymity

Anonmyzation and linkage attacks

- Famous example:
  - After an insurance group released health records which had been stripped of personal information like patient name and address, a CS student was able to “deanonymize” which records belonged to politicians (including the Governor of Massachusetts) by cross referencing with public voter registers.
  - This is an example of a **linkage attack**, where connections to other sources of information work to deanonymize a dataset.
  - Linkage attacks have been successful on a range of anonymized datasets including the Netflix challenge and genome data.



https://arstechnica.com/tech-policy/2009/09/your-secrets-live-online-in-databases-of-ruin/

k-anonymity

- Unfortunately, k-anonymity isn't sufficient for anything but very large datasets with only small numbers of simple fields for each record.
- Intuitively, the more fields and the more possible entries there are in those fields, the more unique a record can be and the harder it is to ensure that there are k equivalent records.

k-anonymity: Example

Original Database to Disclose

	IDENTIFYING VARIABLE	QUASI-IDENTIFIERS		
	ID	Name	Gender	Year of Birth	Test Result
1	1	John Smith	Male	1959	+ve
.	2	Alan Smith	Male	1962	-ve
.	3	Alice Brown	Female	1955	-ve
.	4	Hercules Green	Male	1959	-ve
.	5	Alicia Freds	Female	1942	-ve
.	6	Gill Stringer	Female	1975	-ve
.	7	Marie Kirkpatrick	Female	1966	+ve
.	8	Leslie Hall	Female	1987	-ve
.	9	Bill Nash	Male	1975	-ve
.	10	Albert Blackwell	Male	1978	-ve
.	11	Beverly McCulsky	Female	1964	-ve
.	12	Douglas Henry	Male	1959	+ve
.	13	Freda Shields	Female	1975	-ve
s|	14	Fred Thompson	Male	1967	-ve

2-Anonymization

	QUASI-IDENTIFIERS		
	ID	Gender	Decade of Birth	Test Result
1	1	Male	1950-1959	+ve
.	2	Male	1960-1969	-ve
.	4	Male	1950-1959	-ve
.	6	Female	1970-1979	-ve
.	7	Female	1960-1969	+ve
.	9	Male	1970-1979	-ve
.	10	Male	1970-1979	-ve
.	11	Female	1960-1969	-ve
.	12	Male	1950-1959	+ve
.	13	Female	1970-1979	-ve
|s'|	14	Male	1960-1969	-ve

Disclosed (k-Anonymized) Database (ζ)

Matching

Identification Database (Z)

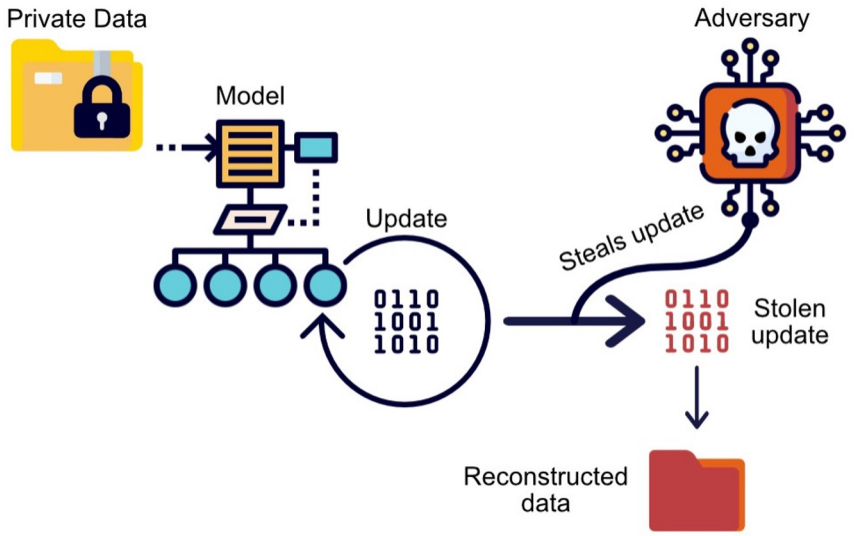
	IDENTIFYING VARIABLE	QUASI-IDENTIFIERS	
	ID	Name	Gender	Year of Birth
1	1	John Smith	Male	1959
.	2	Alan Smith	Male	1962
.	3	Alice Brown	Female	1955
.	4	Hercules Green	Male	1959
.	5	Alicia Freds	Female	1942
.	6	Gill Stringer	Female	1975
.	7	Marie Kirkpatrick	Female	1966
.	8	Leslie Hall	Female	1987
.	9	Bill Nash	Male	1975
.	10	Albert Blackwell	Male	1978
.	11	Beverly McCulsky	Female	1964
.	12	Douglas Henry	Male	1959
.	13	Freda Shields	Female	1975
.	14	Fred Thompson	Male	1967
.	15	Joe Doe	Male	1961
.	16	Mark Fractus	Male	1974
.	17	Lillian Barley	Female	1978
.	18	Jane Doe	Female	1961
.	19	Nina Brown	Female	1968
.	20	William Cooper	Male	1973
.	21	Kathy Last	Female	1966
.	22	Deitmar Plank	Male	1967
.	23	Anderson Hoyt	Male	1971
.	24	Alexandra Knight	Female	1974
.	25	Helene Arnold	Female	1977
.	26	Anderson Heft	Male	1968
.	27	Almond Zipf	Male	1954
.	28	Alex Long	Female	1952
.	29	Britney Goldman	Female	1956
.	30	Lisa Marie	Female	1988
|U|	31	Natasha Markhov	Female	1941

K. El Emam et al. Protecting privacy using k-anonymity. J Am Med Inform Assoc. 2008
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2528029/

Privacy attacks: Model inversion

- Attacker uses internal representations of the joint model to reconstruct individual training samples or their sensitive attributes

- Example:
  - Inversion of training data in collaborative pneumonia classification

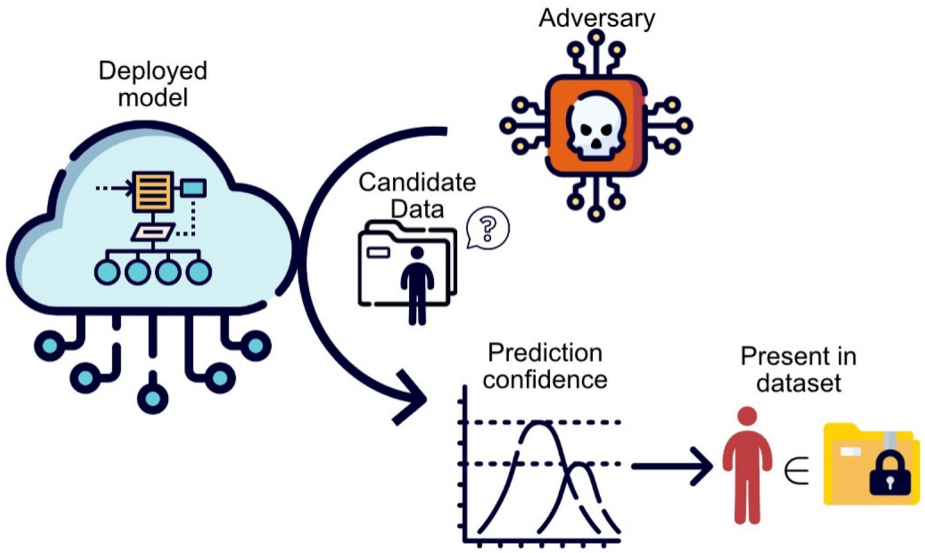


Usynin et al. 2021, Adversarial interference and its mitigations in privacy-preserving collaborative machine learning, *Nature Machine Intelligence*

Privacy attacks: Membership inference

- Attacker obtains a data record and determines if it was used to train a particular model

- Example:
  - Determining if a specific patient was part of the HIV-positive dataset

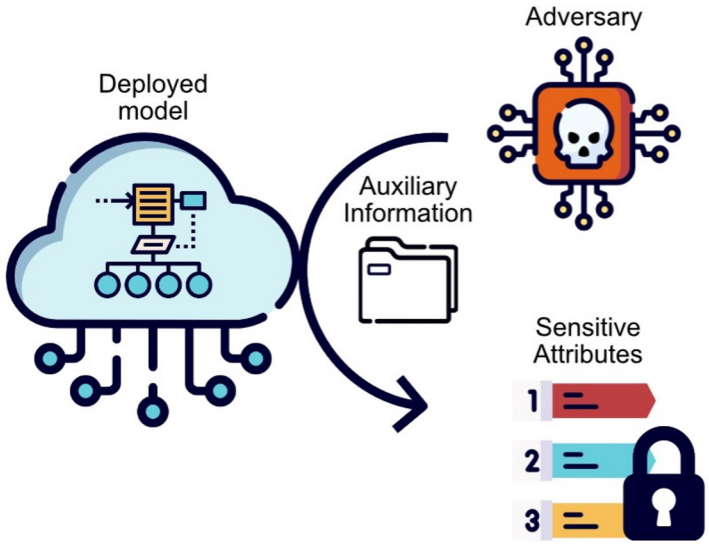


Usynin et al. 2021, Adversarial interference and its mitigations in privacy-preserving collaborative machine learning, *Nature Machine Intelligence*

Privacy attacks: Attribute inference

- Attacker uses model access and auxiliary information about the victim to obtain the sensitive values of their data

- Example:
  - Given access to a model trained on patient records and a specific patient's public information infer their HIV status



Usynin et al. 2021, Adversarial interference and its mitigations in privacy-preserving collaborative machine learning, *Nature Machine Intelligence*

Differential Privacy

Randomized responses

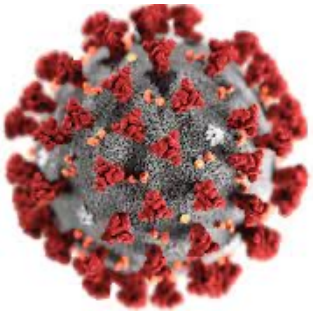
- Enables draw statistical conclusion from datasets without revealing information about individual data points.

Differential Privacy

Randomized responses

- Enables draw statistical conclusion from datasets without revealing information about individual data points.
- Realised by adding a controlled amount of noise

Differential Privacy: Randomized responses



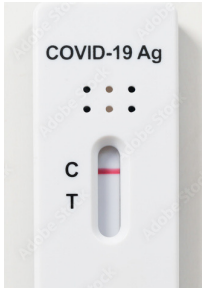
Differential Privacy: Randomized responses

Head: Answer truthfully

Tail: Second coin flip

Head

Tail



True signal

Noise

Observed signal

Differential Privacy: Randomized responses

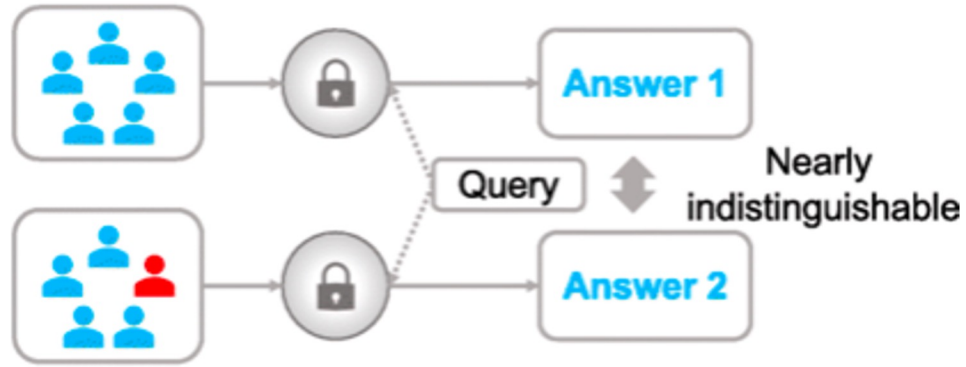
- In this example, there is a parameter which is the probability that the true response is recorded:
  - If it's very likely that the true response is recorded, then there is less privacy protection.
  - Conversely, if it's unlikely that the true response is recorded, then there is more.
  - It's also clear that, regardless of the probability, if an individual is surveyed multiple times, then there will be less protection, even if their answer is potentially randomized every time.

- Differential privacy formalizes how we define, measure and track the privacy protection afforded to an individual as functions of factors like randomization probabilities and number of times surveyed.

Differential privacy

- An algorithm can be made approximately invariant to inclusion of a single data sample

- This is achieved through the addition algorithm



Differential privacy (continued)

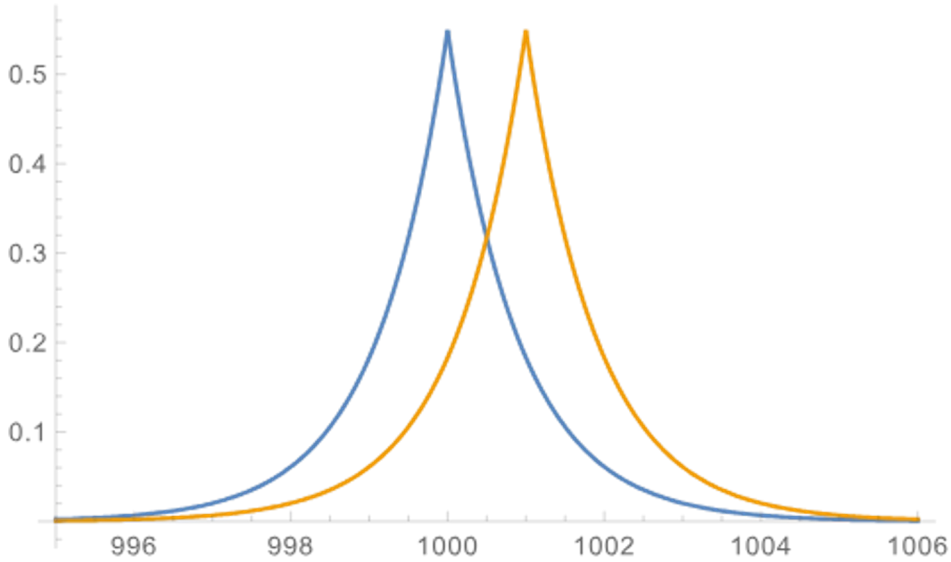
- This can be mathematically defined using the following (relaxed) expression:

A process A is ε-differentially private if for all databases D_1 and D_2 which differ in only one individual:

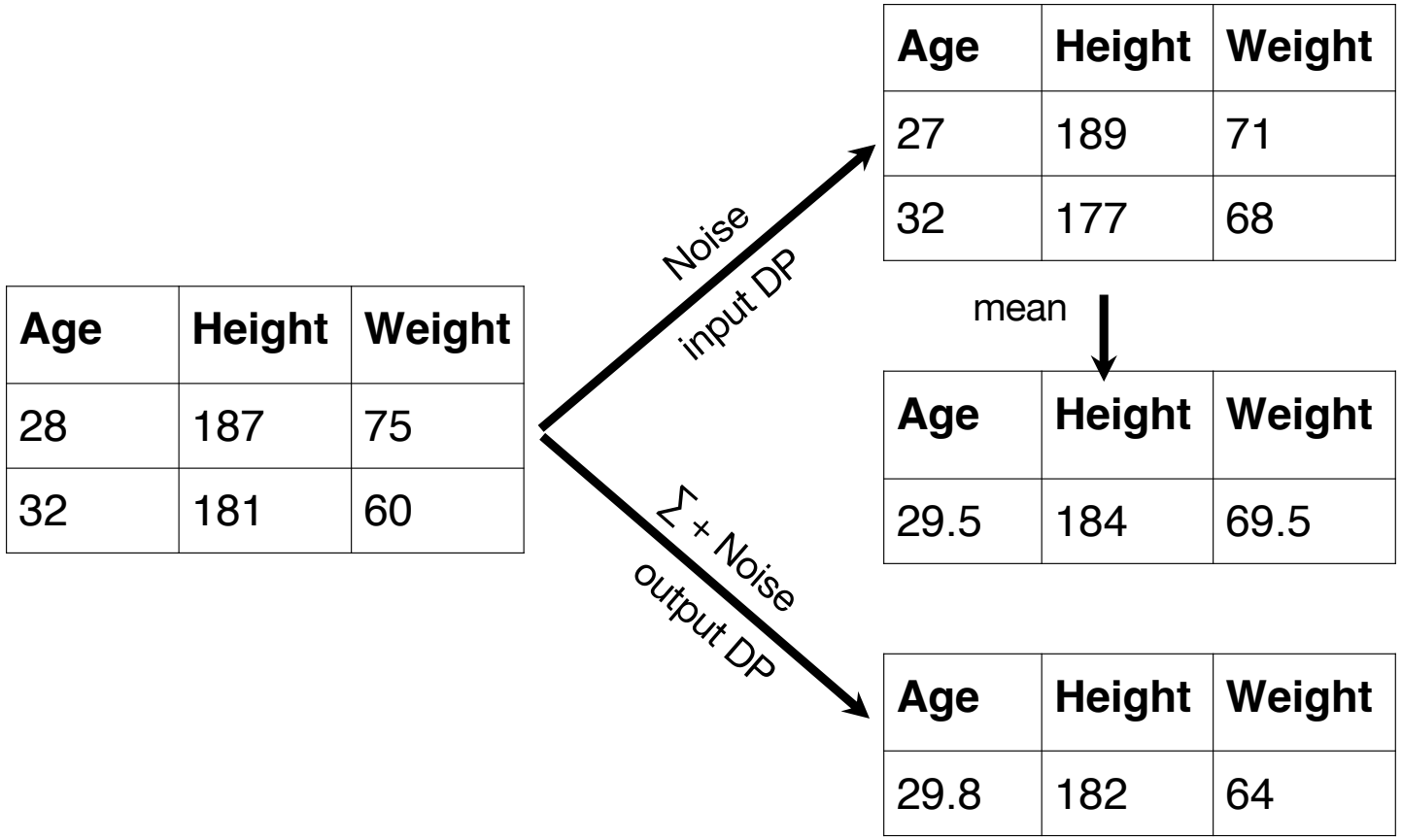
$$\mathbb{P}[A(D_1) = O] \leq e^{\varepsilon} \cdot \mathbb{P}[A(D_2) = O]$$

Differential privacy (continued)

- This is done through the addition of noise to the output of the query so then the results in two datasets look approximately like this:



Differential privacy (continued)



Concrete mitigation: Differentially private stochastic gradient descent (DP-SGD)

1. Compute gradients for each individual sample (they represent independent clients)
2. Clip the calculated gradients to obtain a known sensitivity
3. Add the noise scaled by the sensitivity from step 2
4. Perform the gradient descent step

Algorithm 1 Differentially private SGD (Outline)

Input: Examples $\{x_1, \ldots, x_N\}$, loss function $\mathcal{L}(\theta) = \frac{1}{N}\sum_i \mathcal{L}(\theta, x_i)$. Parameters: learning rate η_t, noise scale σ, group size L, gradient norm bound C.

Initialize θ_0 randomly

for $t \in [T]$ **do**

Take a random sample L_t with sampling probability L/N

Compute gradient

For each $i \in L_t$, compute $\mathbf{g}_t(x_i) \leftarrow \nabla_{\theta_t}\mathcal{L}(\theta_t, x_i)$

Clip gradient

$\bar{\mathbf{g}}_t(x_i) \leftarrow \mathbf{g}_t(x_i) / \max\left(1, \frac{\|\mathbf{g}_t(x_i)\|_2}{C}\right)$

Add noise

$\tilde{\mathbf{g}}_t \leftarrow \frac{1}{L}\sum_i (\bar{\mathbf{g}}_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I}))$

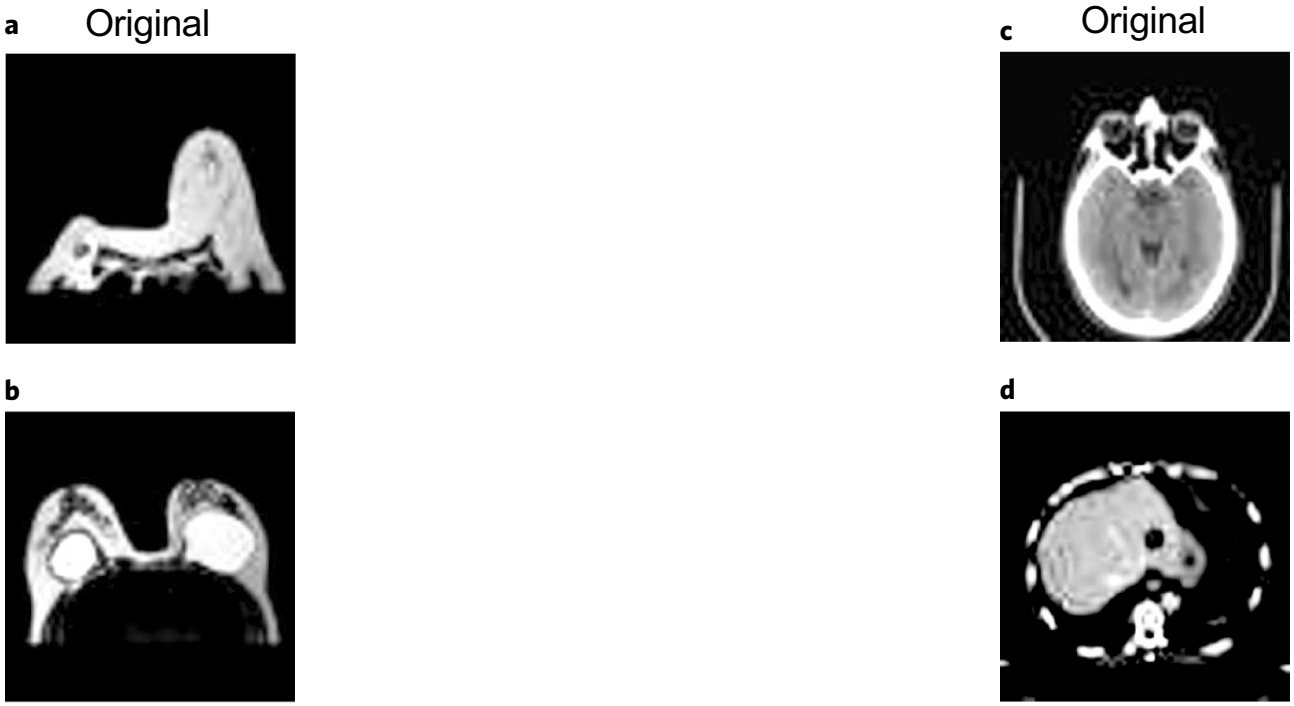
Descent

$\theta_{t+1} \leftarrow \theta_t - \eta_t \tilde{\mathbf{g}}_t$

Output θ_T and compute the overall privacy cost (ε, δ) using a privacy accounting method.

Abadi, Martin, et al. "Deep learning with differential privacy." *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*. 2016.

Concrete mitigation: Differentially private stochastic gradient descent (DP-SGD)



Kaissis et al. Nature Machine Intelligence, 2021

Imperial College London

Imperial College London

Interpretability and Explainability

Daniel Rueckert
Department of Computing
Imperial College London, UK

Trustworthy AI/ML

Seven key requirements for trustworthy AI/ML

https://ec.europa.eu/digital-single-market/en/news/ethics-guidelines-trustworthy-ai

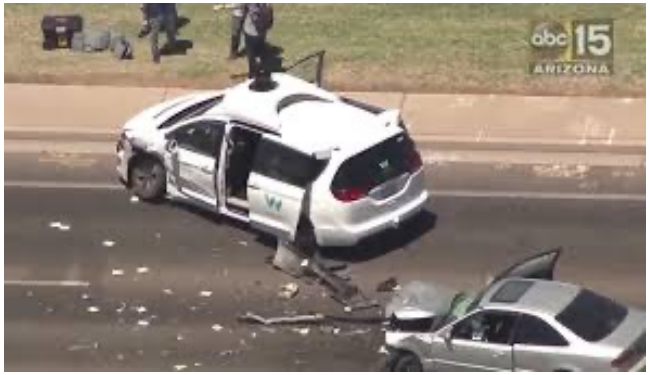
Interpretability: Why is this important?

- This is not a new problem, so why now?
  - Complexity and prevalence!
  - Safety and robustness is critical
  - Generating knowledge

Interpretability: Why is this important?

Safety

Science



Robustness

Interpretability: Why is this important?

- This is not a new problem, so why now?
  - Complexity and prevalence!
  - Safety is critical
  - Generating knowledge

- Debugging machine learning
  - Why does my model not train?
  - Why does my model exhibit unexpected/unintuitive behaviour?

Interpretability: Why is this important?

- Debugging machine learning models

Data during training

Data during deployment

Interpretability: Why is this important?

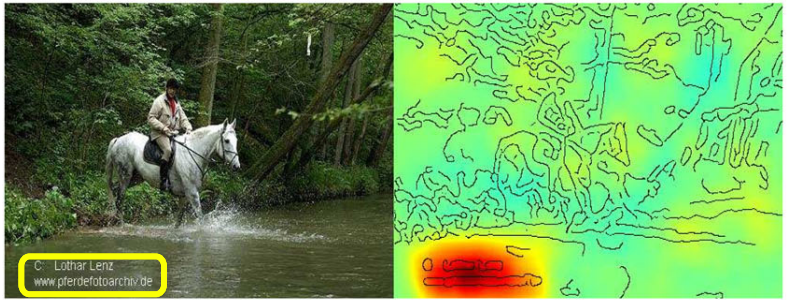
- What have we learnt?

ARTICLE
https://doi.org/10.1038/s41467-019-08987-4 OPEN

Unmasking Clever Hans predictors and assessing what machines really learn

Sebastian Lapuschkin[1], Stephan Wäldchen[2], Alexander Binder[3], Grégoire Montavon[2], Wojciech Samek[1] & Klaus-Robert Müller[2,4,5]

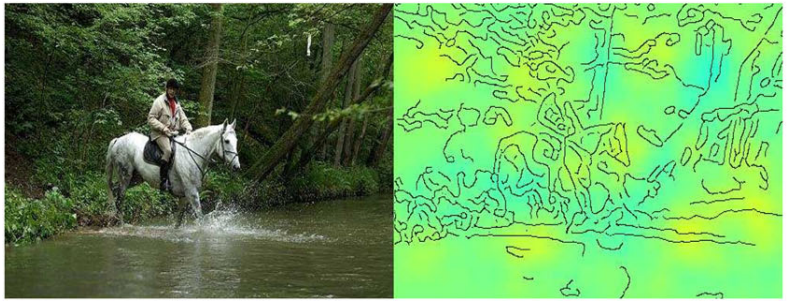
Horse-picture from Pascal VOC data set



Source tag present

↓

Classified as horse



No source tag present

↓

Not classified as horse

Artificial picture of a car

Interpretability: Why is this important?

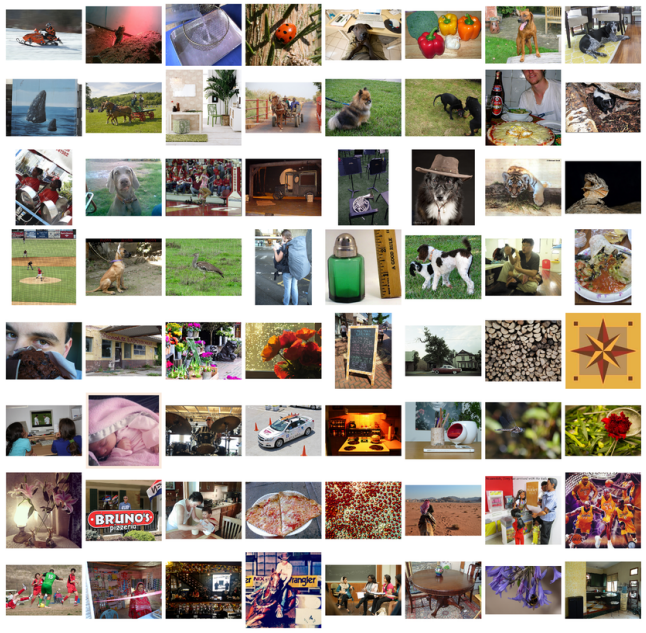
- This is not a new problem, so why now?
  - Complexity and prevalence!
  - Safety is critical
  - Generating knowledge

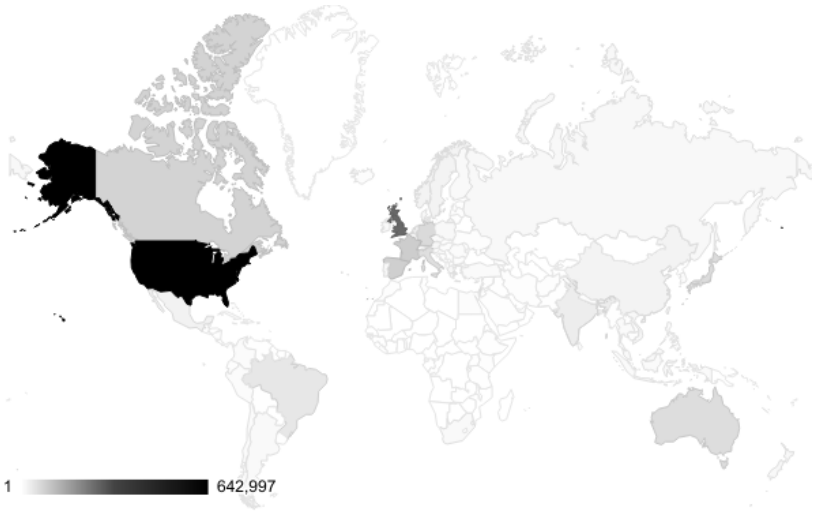
- Debugging machine learning
  - Why does my model not train?
  - Why does my model exhibit unexpected/unintuitive behaviour?

- To use machine learning responsibly we need to ensure that
  - Our values are aligned
  - Our knowledge is reflected

Interpretability: Why is this important?

- Detecting bias

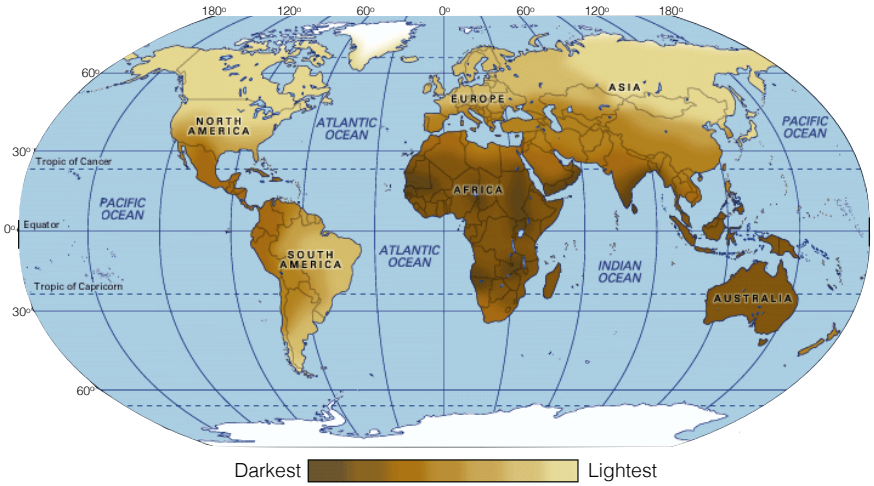


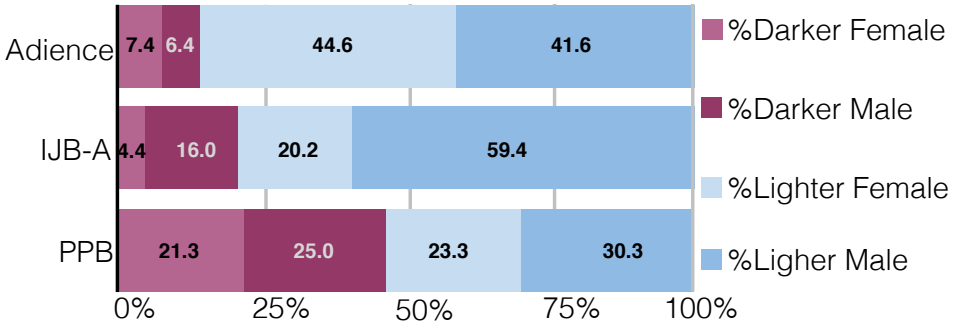


Geographical distribution of images from the ImageNet dataset

Interpretability: Why is this important?

- Detecting bias



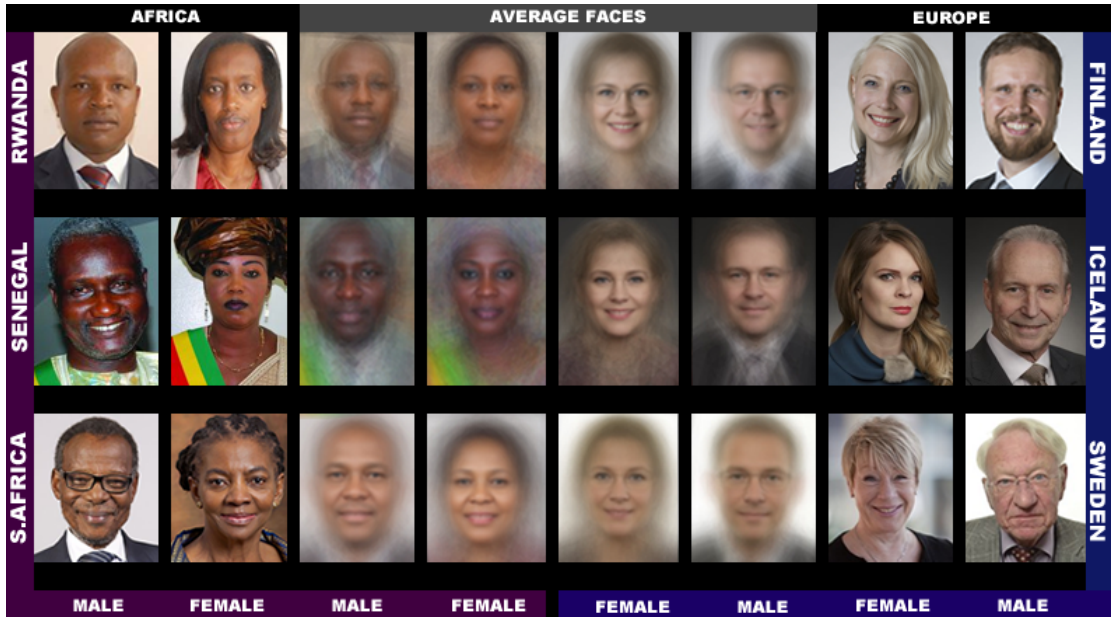


Globe image from Encyclopedia Britannica

Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification, J. Buolamwini & T. Gebru, PMLR 81:77-91, 2018.

Interpretability: Why is this important?

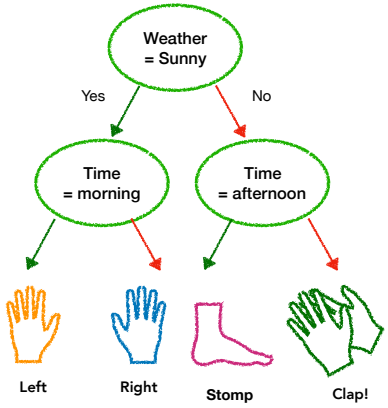
- Detecting bias

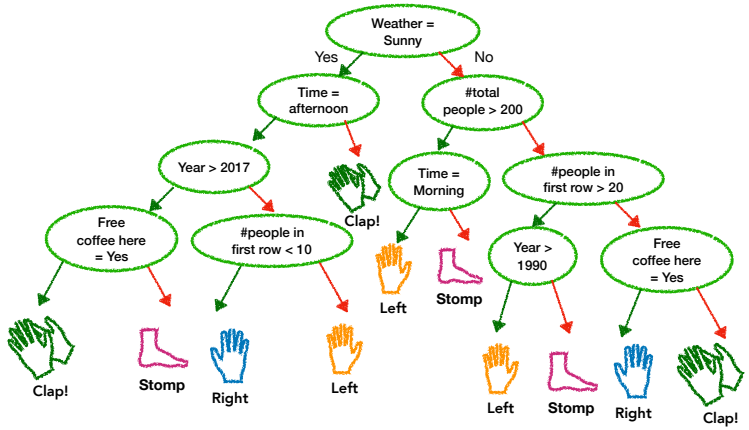


Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification, J. Buolamwini & T. Gebru, PMLR 81:77-91, 2018.

Interpretability: Common misunderstandings

- Simple ML models (e.g. linear models or decision trees are interpretable)





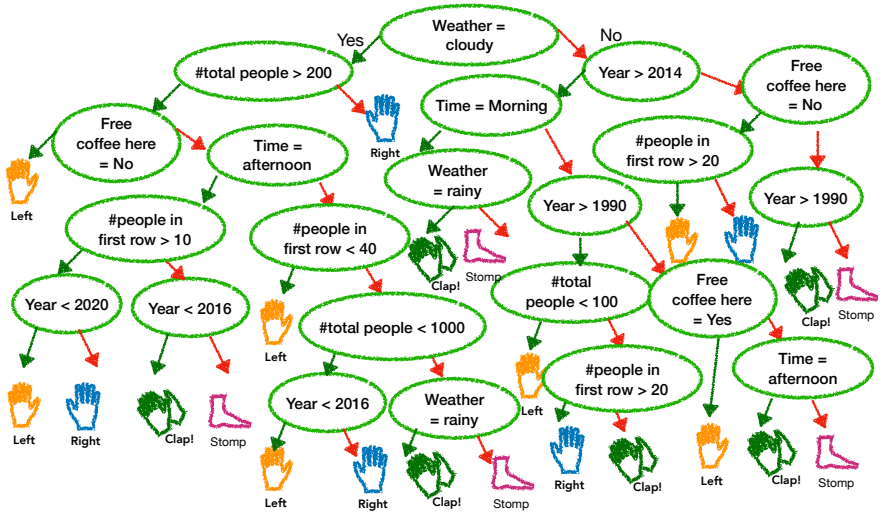
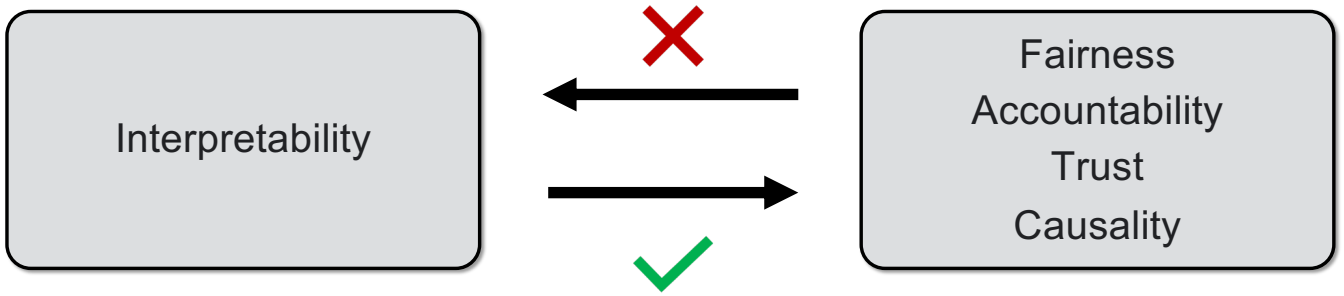


Figure by Been Kim, Google, CVPR 2018

Interpretability: Common misunderstandings

- Trust, fairness and interpretability are all the same thing

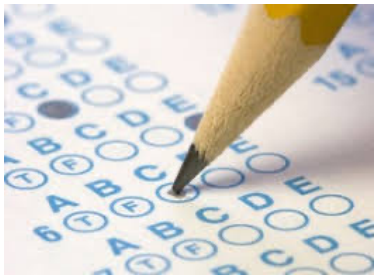


– Interpretability can help to formalize concepts such as fairness or trust
– Once formalized it may not be need anymore

Adapted from slides by Been Kim, Google, CVPR 2018

Interpretability: Common misunderstandings

- We don’t always care about interpretability:
  - No significant consequences or when predictions are all you need
  - Sufficiently well-studied problem
  - Prevent gaming the system



Interpretability: Types of methods

- Before building ML model
- During building ML model
- After building ML model

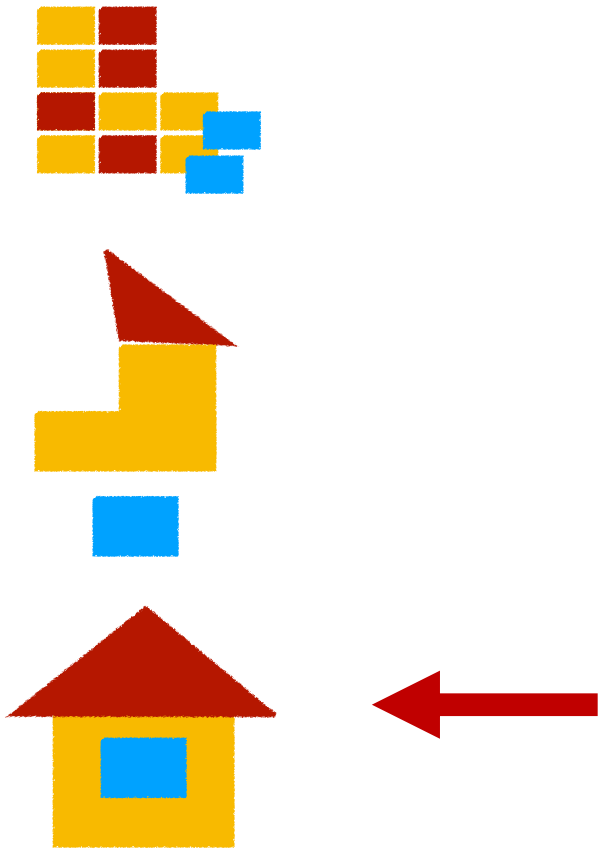


Figure by Been Kim, Google, CVPR 2018

How can we interpret an existing ML model?

- Ablation test: How important is a data point or feature?
  - Train without certain data or features and observe/study the impact
  - Difficult and expensive

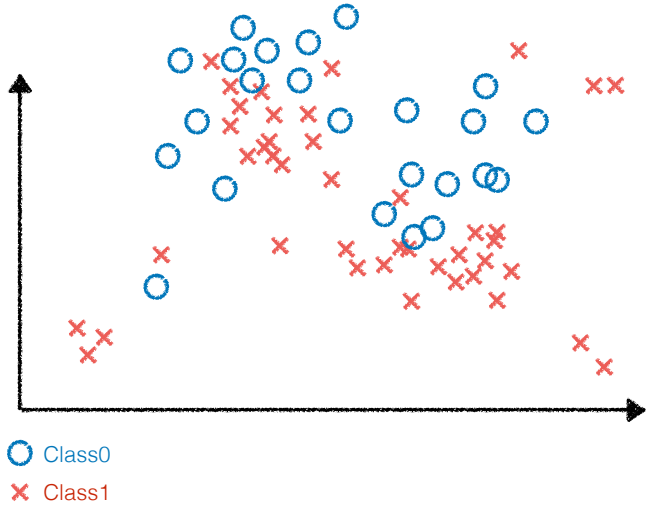


Figure by Been Kim, Google, CVPR 2018

How can we interpret an existing ML model?

- Ablation test: How important is a data point or feature?
  - Train without certain data or features and observe/study the impact
  - Difficult and expensive

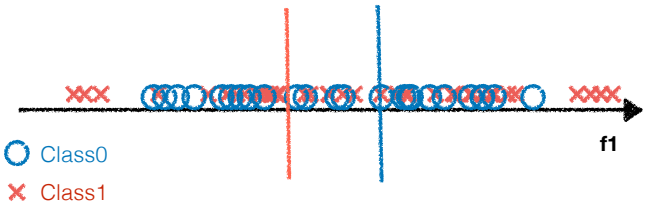


Figure by Been Kim, Google, CVPR 2018

How can we interpret an existing ML model?

- Fit functions (use first derivatives)
  - Sensitivity analysis
  - Saliency maps

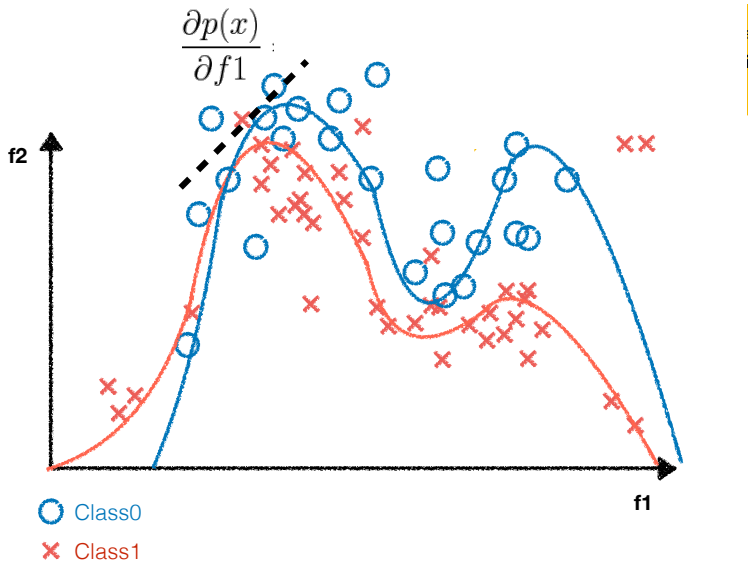


Figure by Been Kim, Google, CVPR 2018

How can we interpret an existing ML model?

- Fit functions (use first derivatives)
  - Sensitivity analysis
  - Saliency maps

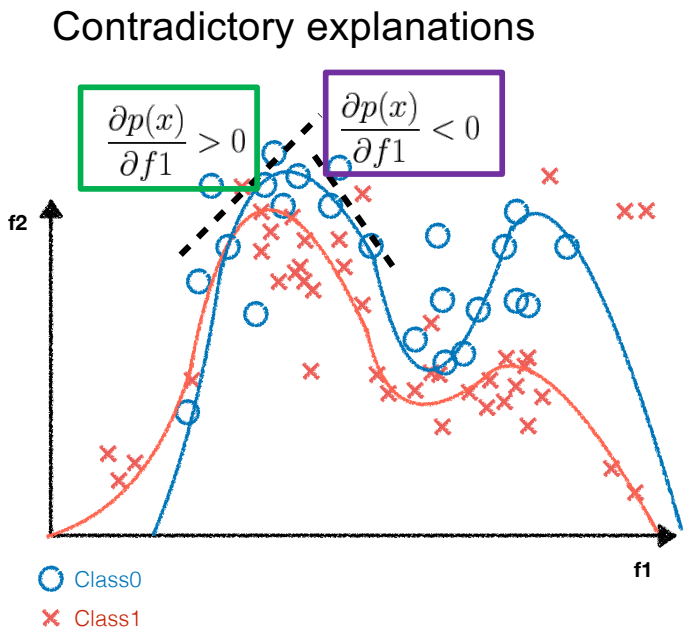


Figure by Been Kim, Google, CVPR 2018

How can we interpret an existing ML model?

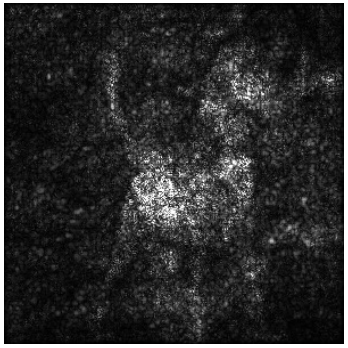
A trained
machine learning model
(e.g. neural network)

$p(z)$

"dogness"

Why is this a
dog?

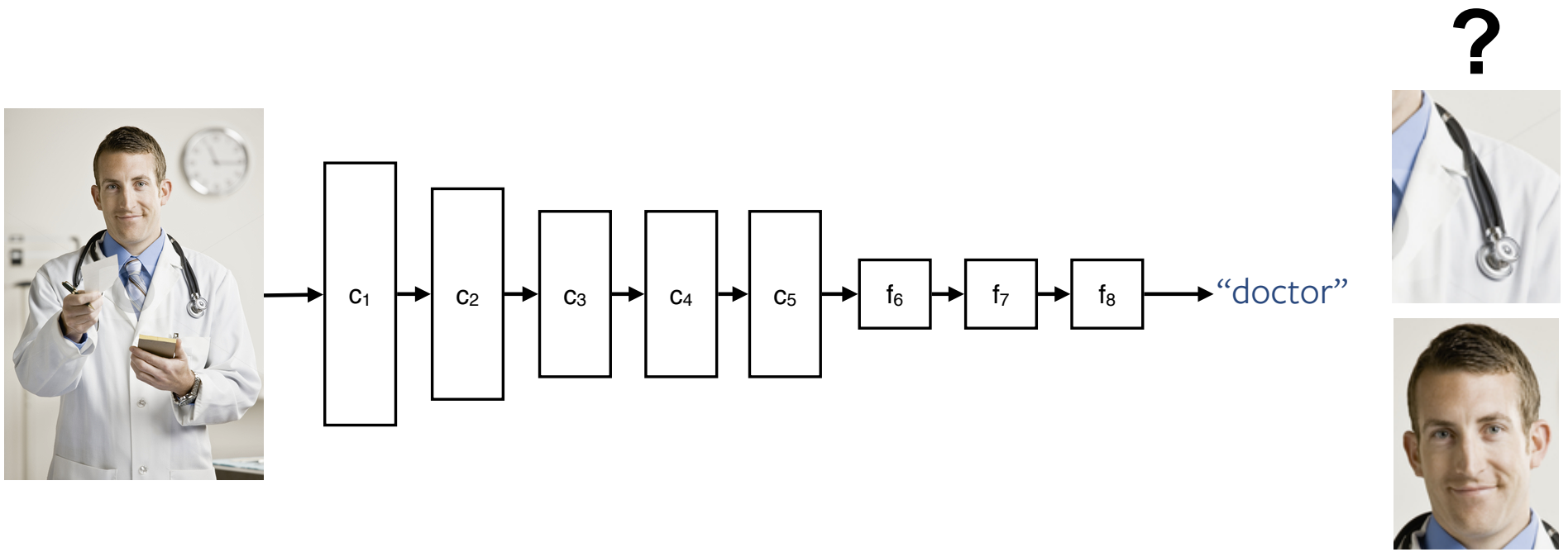
How can we interpret an existing ML model?



$$\frac{\partial p(z)}{\partial x_{i,j}}$$

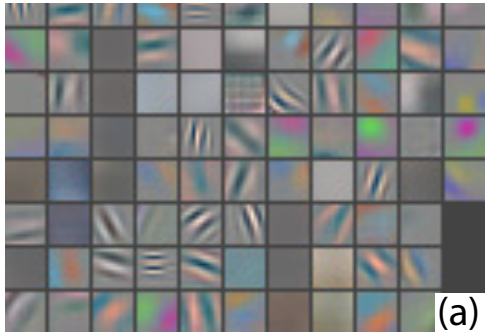
How can we interpret an existing ML model?

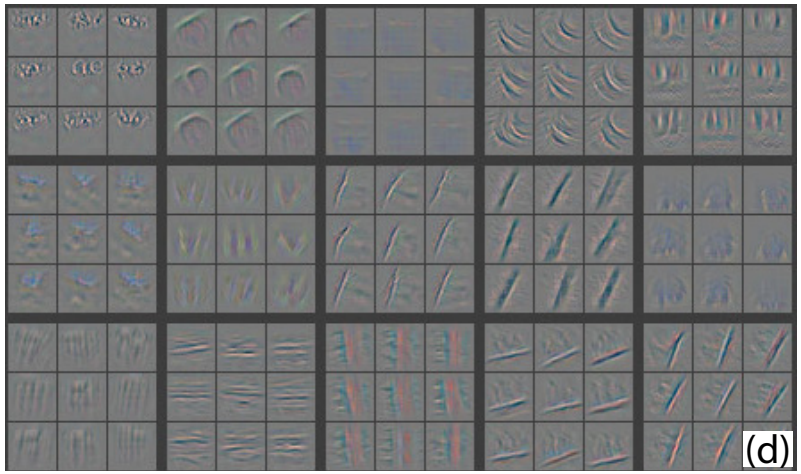
- Visualization and attribution:
  - Identify input features responsible for model decision



How can we interpret an existing ML model?

- Direct visualization of filters
- Easy to implement
- Limited practical value
  - First layers are easy to interpret (mostly low-level features)
  - Higher layers are more difficult to interpret (non-interpretable features)

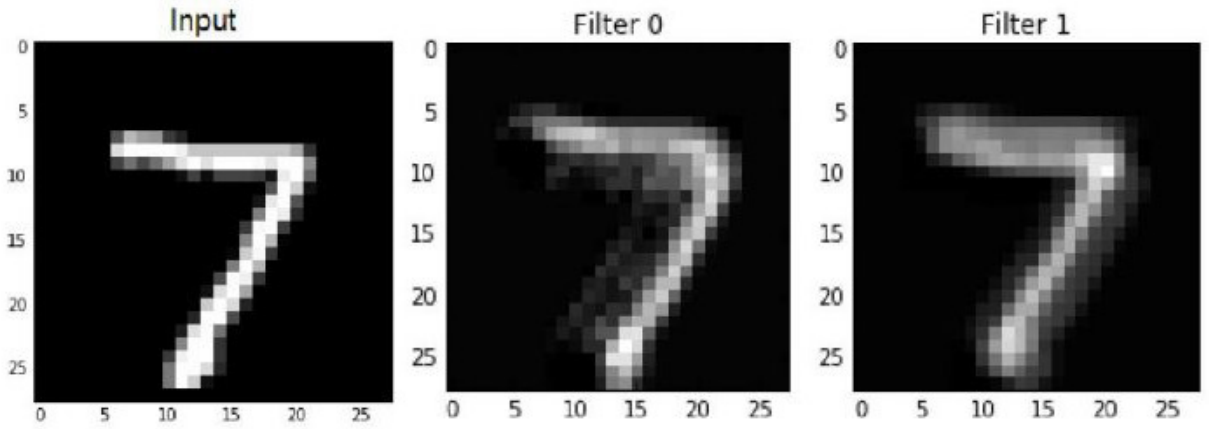




https://arxiv.org/abs/1311.2901

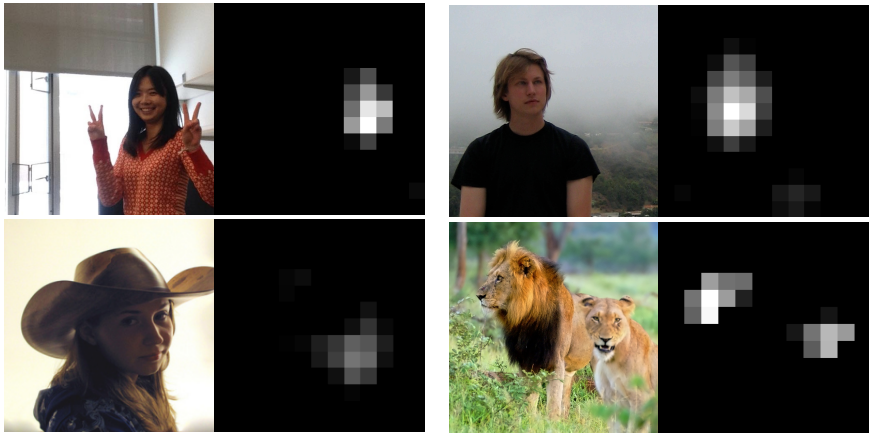
How can we interpret an existing ML model?

- Problem: Visualization of filters has limited value
- Solution: Instead visualize activations generated by kernels
  - Strong response: Feature is present
  - Weak response: Feature is not present
  - Easy to implement
  - Easy to interpret for early layers



How can we interpret an existing ML model?

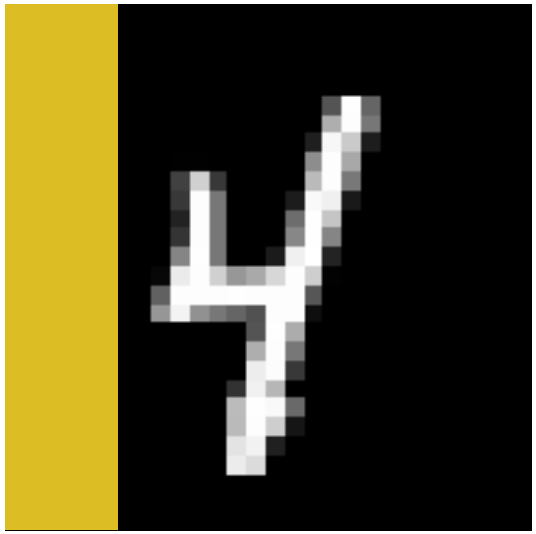
- Problem: Visualization of filters has limited value
- Solution: Instead visualize activations generated by kernels
  - Strong response: Feature is present
  - Weak response: Feature is not present
  - Easy to implement
  - Easy to interpret for early layers
  - Higher layers are more sparse
  - Channels may correspond to specific features

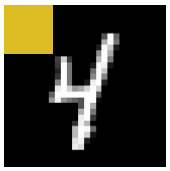


https://arxiv.org/abs/1506.06579

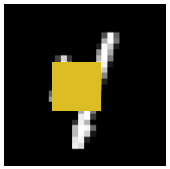
How can we interpret an existing ML model?
Occlusions

- Idea: Mask out region in the input image and observe network output
- If masked out region causes a significant drop in confidence, the masked-out region is important

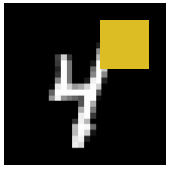




number 4 = 0.97



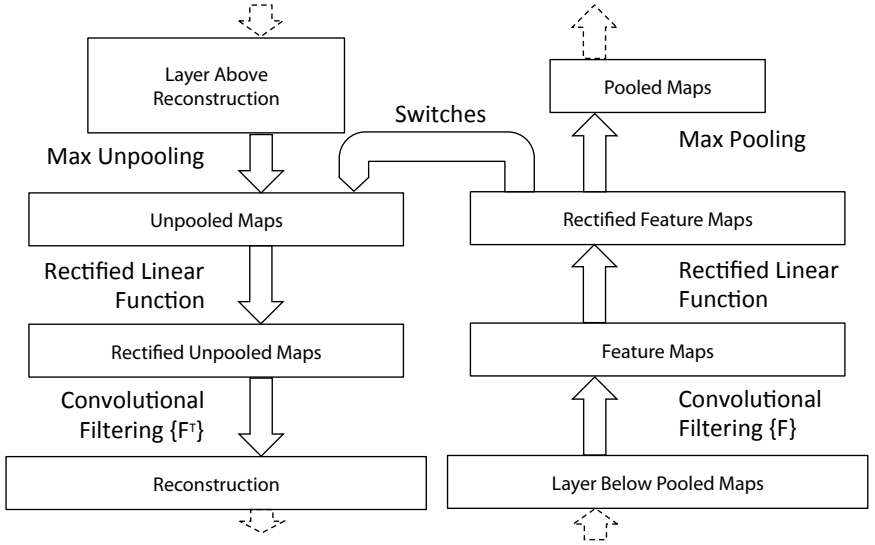
number 4 = 0.58



number 4 = 0.91

How can we interpret an existing ML model?
Saliency maps

- **DeconvNet**
  - Given a trained network and an image
  - Chose activation at one layer (set all others to zero)
  - Invert network

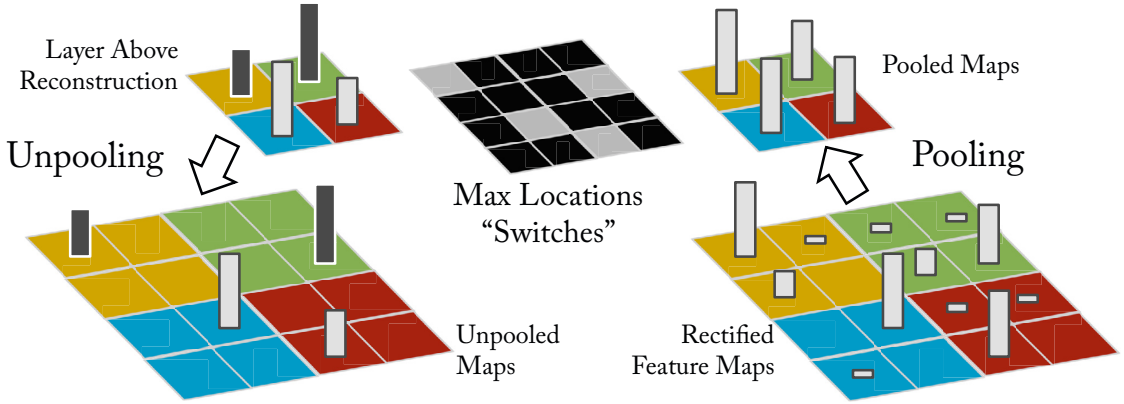


  - No training involved
  - Backward pass in network is almost identical to backpropagation (apart from ReLUs)

Visualizing and Understanding Convolutional Networks, M. D. Zeiler and R. Fergus, ECCV 2014

How can we interpret an existing ML model?
Saliency maps

- **DeconvNet**
  - Given a trained network and an image
  - Chose activation at one layer (set all others to zero)
  - Invert network

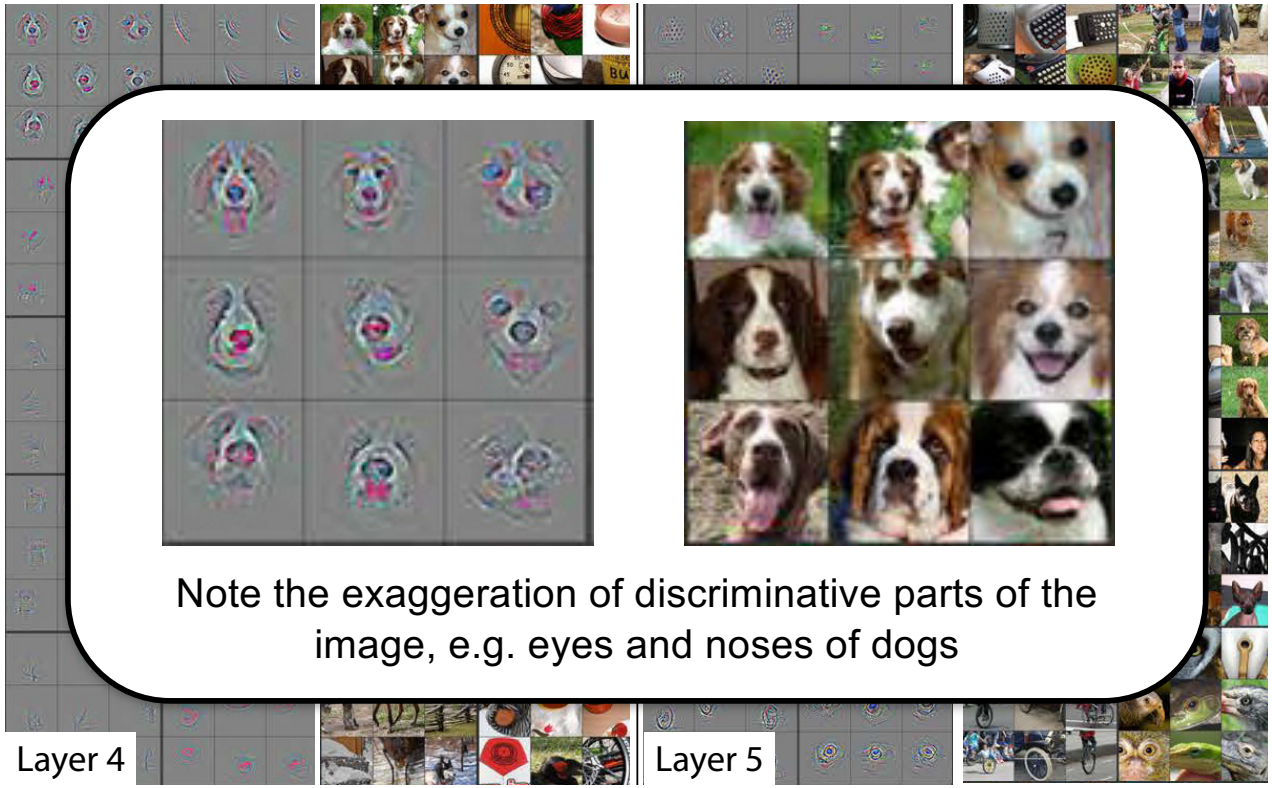


  - No training involved
  - Backward pass in network is almost identical to backpropagation (apart from ReLUs)

Visualizing and Understanding Convolutional Networks, M. D. Zeiler and R. Fergus, ECCV 2014

How can we interpret an existing ML model?
Saliency maps

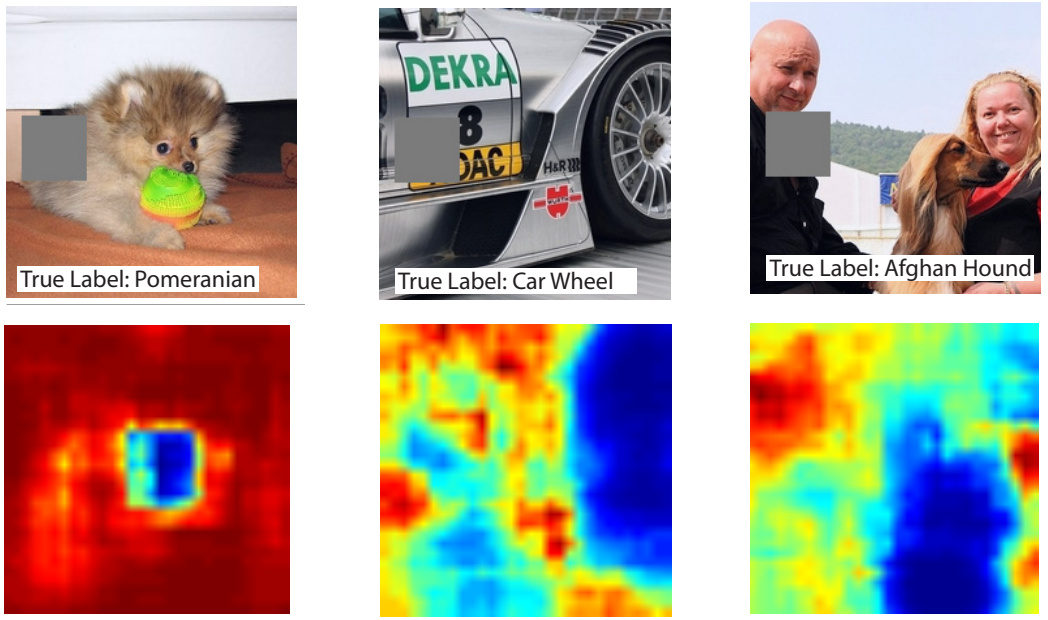
- **DeconvNet**



Visualizing and Understanding Convolutional Networks, M. D. Zeiler and R. Fergus, ECCV 2014

How can we interpret an existing ML model?
Occlusions

- Idea: Mask out region in the input image and observe network output
- If masked out region causes a significant drop in confidence, the masked-out region is important

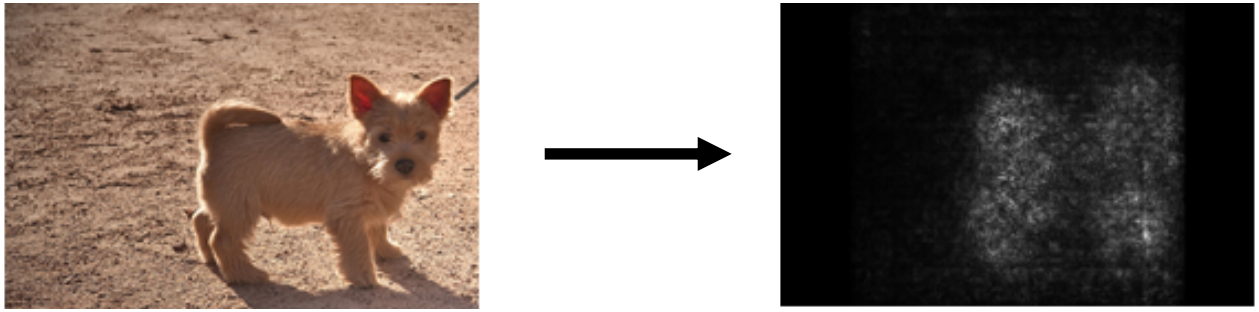


Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps, K. Simonyan, A. Vedaldi, and A. Zisserman ICLR, 2014.

How can we interpret an existing ML model?
Saliency maps

- Question:
  - Which pixels are most significant to a neuron?
  - How would they need to change to most affect the activation of the neuron?
- Solution:
  - Use back propagation but differentiate activation with respect to input pixels, not weights

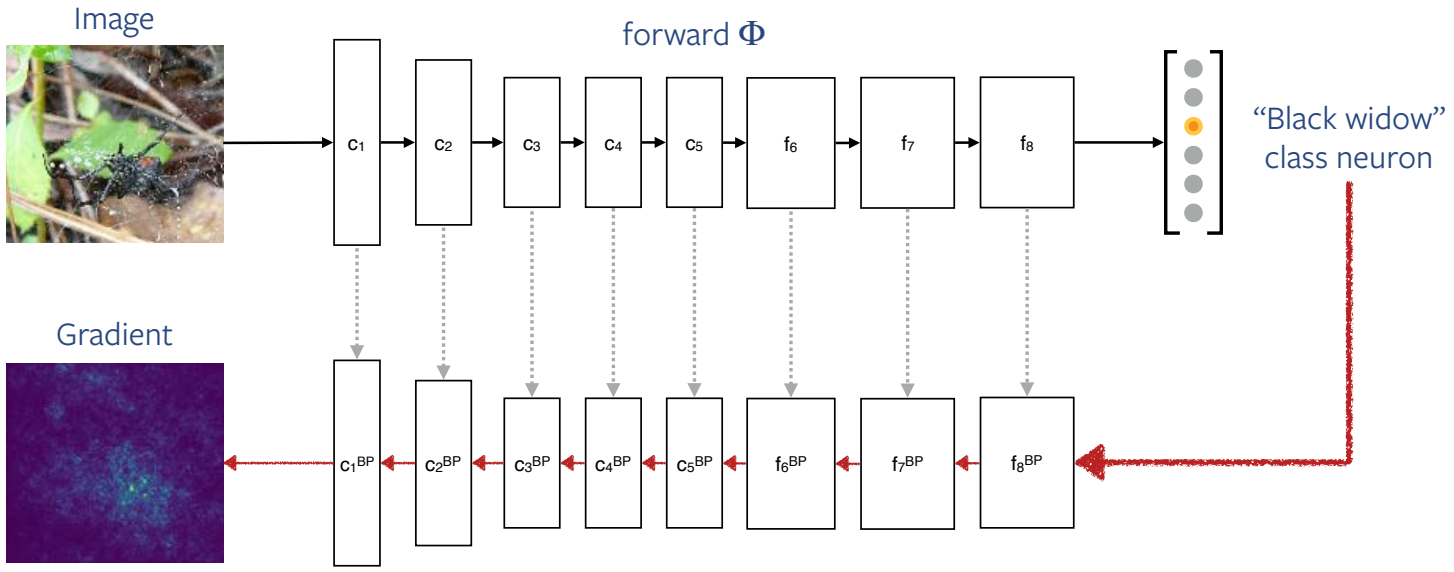
weights of the network are fixed $\frac{\partial h}{\partial x_i}$ ~~$\frac{\partial f}{\partial \theta_i}$~~



Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps, K. Simonyan, A. Vedaldi, and A. Zisserman ICLR, 2014.

How can we interpret an existing ML model?
Saliency maps

- **Gradient (backprogagation)**
  – Define loss as activation of arbitrary neuron in any layer (last layer is most interesting)



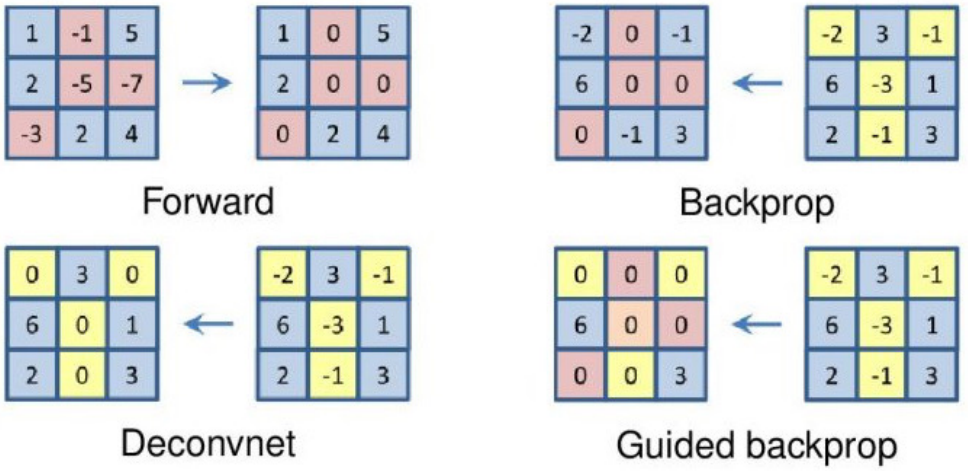
backward pass = $\frac{\partial h}{\partial x_i}$ Salient pixels light up

Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps, K. Simonyan, A. Vedaldi, and A. Zisserman ICLR, 2014.

How can we interpret an existing ML model?
Saliency maps

- **Guided backpropagation**
  - Improve results by “guiding” the backpropagation process
  - Idea:
    - Positive gradients = features the neuron is interested in
    - Negative gradients = features the neuron is not interested in

  ➜ Set all negative gradients in the backpropagation to zero
  - Propagating through the ReLU:



How can we interpret an existing ML model?
Saliency maps

Deconvolution

Visualizing and understanding convolutional neural networks
Zeiler & Fergus, ECCV, 2014

Gradient (backpropagation)

Deep inside convolutional networks: Visualising image classification models and saliency maps
Simonyan, Vedaldi, Zisserman, ICLR 2014

Guided backpropagation

Striving for simplicity: The all convolutional net
Springenberg, Dosovitsky, Brox, Riedmiller, ICLR, 2015

How can we interpret an existing ML model?
Saliency maps

Deconvolution

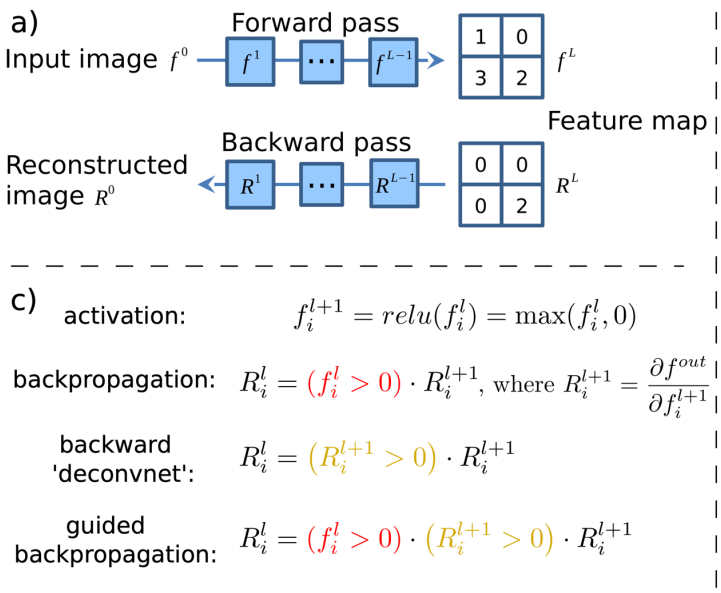
Visualizing and understanding convolutional neural networks
Zeiler & Fergus, ECCV, 2014

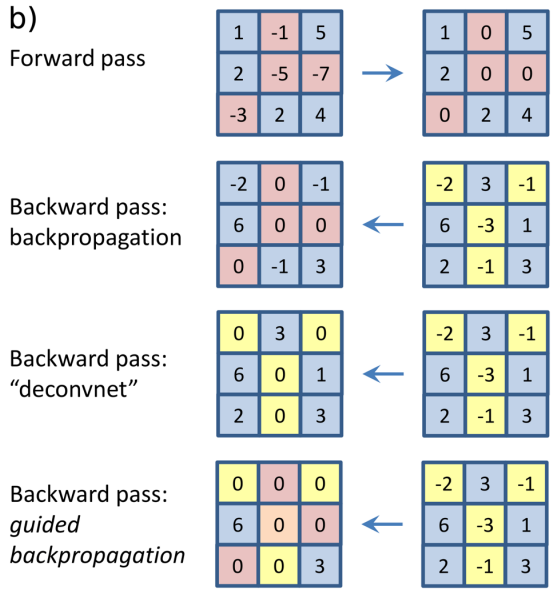
Gradient (backpropagation)

Deep inside convolutional networks: Visualising image classification models and saliency maps
Simonyan, Vedaldi, Zisserman, ICLR 2014

Guided backpropagation

Striving for simplicity: The all convolutional net
Springenberg, Dosovitsky, Brox, Riedmiller, ICLR, 2015





How can we interpret an existing ML model?
Saliency maps

Deconvolution

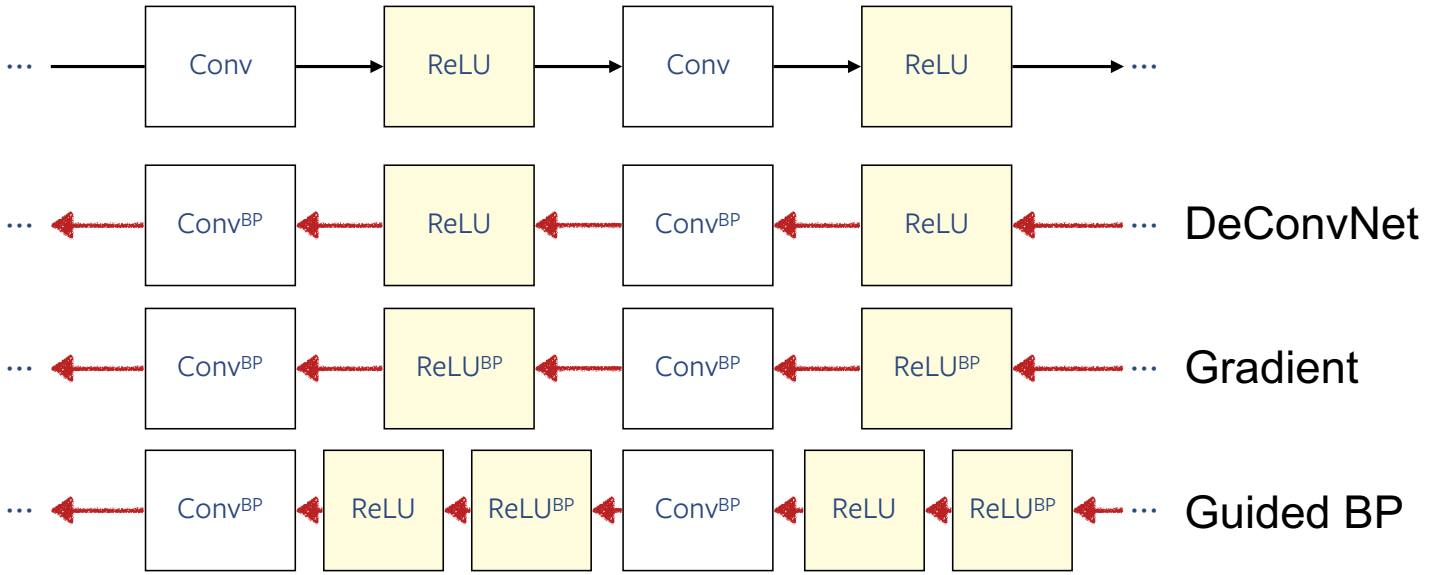
Visualizing and understanding convolutional neural networks
Zeiler & Fergus, ECCV, 2014

Gradient (backpropagation)

Deep inside convolutional networks: Visualising image classification models and saliency maps
Simonyan, Vedaldi, Zisserman, ICLR 2014

Guided backpropagation

Striving for simplicity: The all convolutional net
Springenberg, Dosovitsky, Brox, Riedmiller, ICLR, 2015



How can we interpret an existing ML model?
Saliency maps

Deconvolution

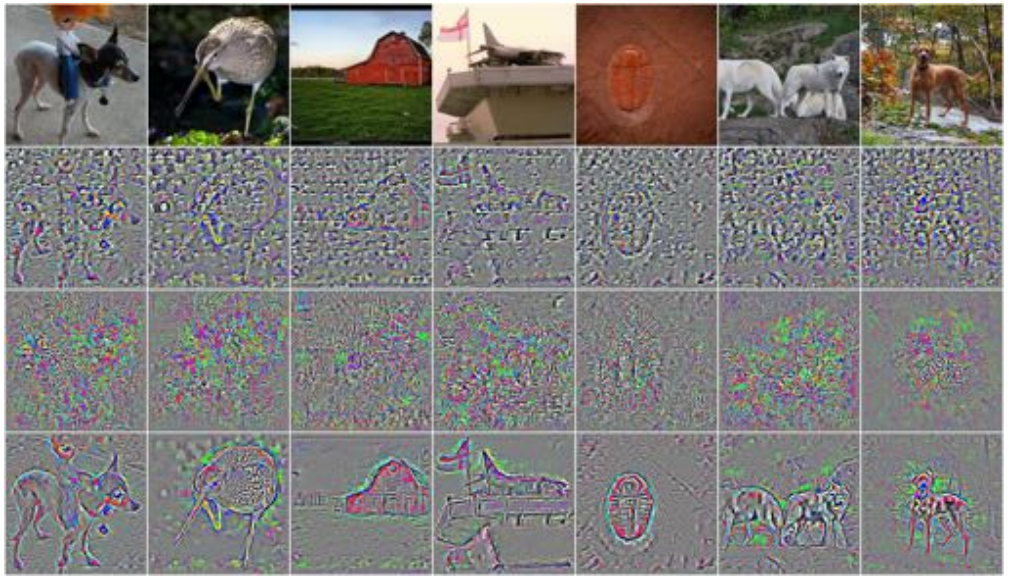
Visualizing and understanding convolutional neural networks
Zeiler & Fergus, ECCV, 2014

Gradient (backprogagation)

Deep inside convolutional networks: Visualising image classification models and saliency maps
Simonyan, Vedaldi, Zisserman, ICLR 2014

Guided backprogagation

Striving for simplicity: The all convolutional net
Springenberg, Dosovitsky, Brox, Riedmiller, ICLR, 2015



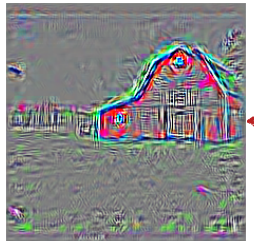
DeConvNet

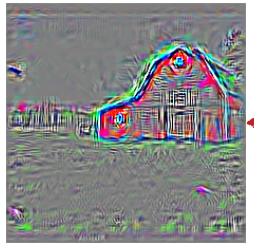
Gradient

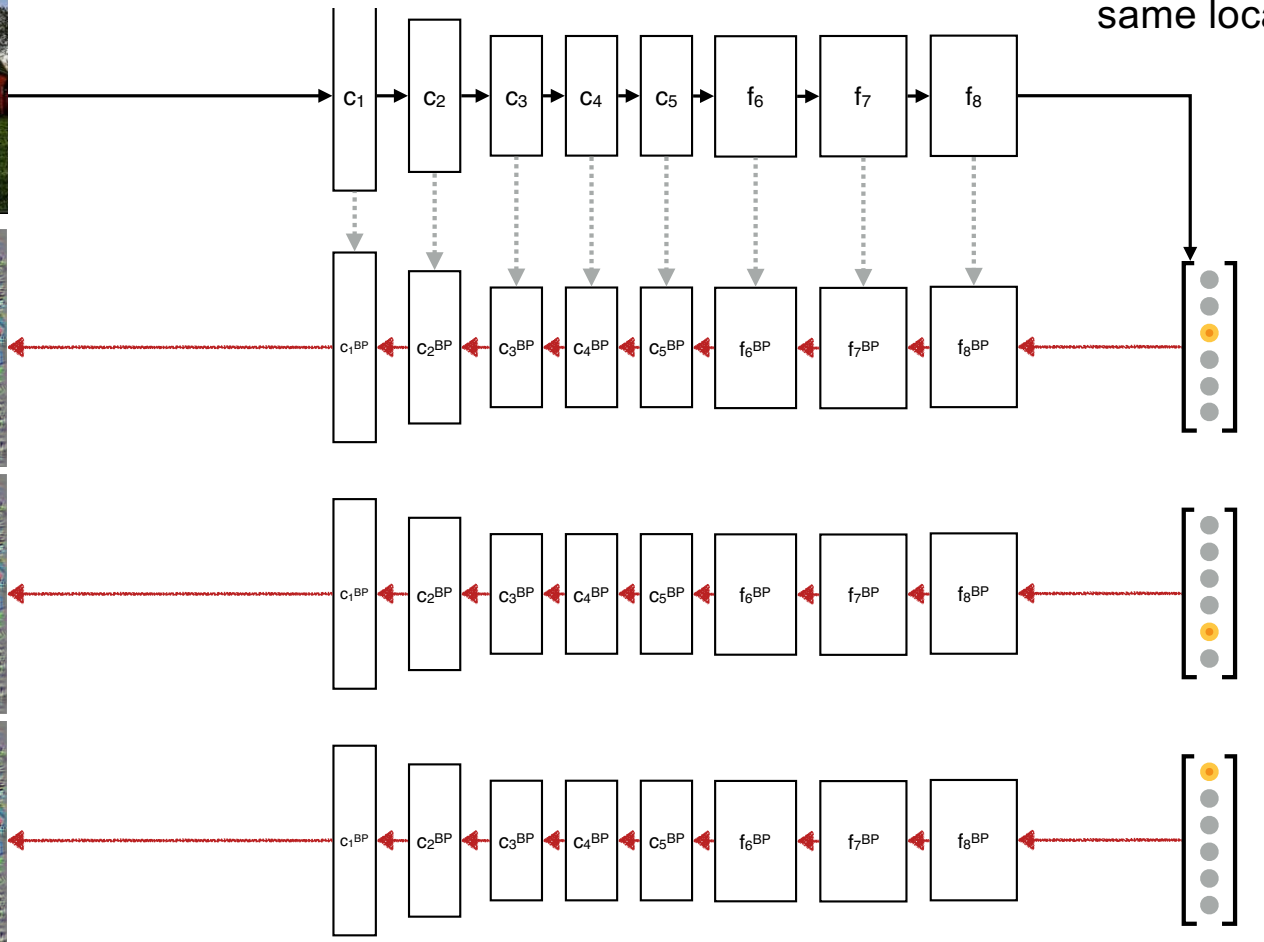
Guided BP

Lack of channel specificity

The saliency of neurons at the same location is about the same







Saliency for:

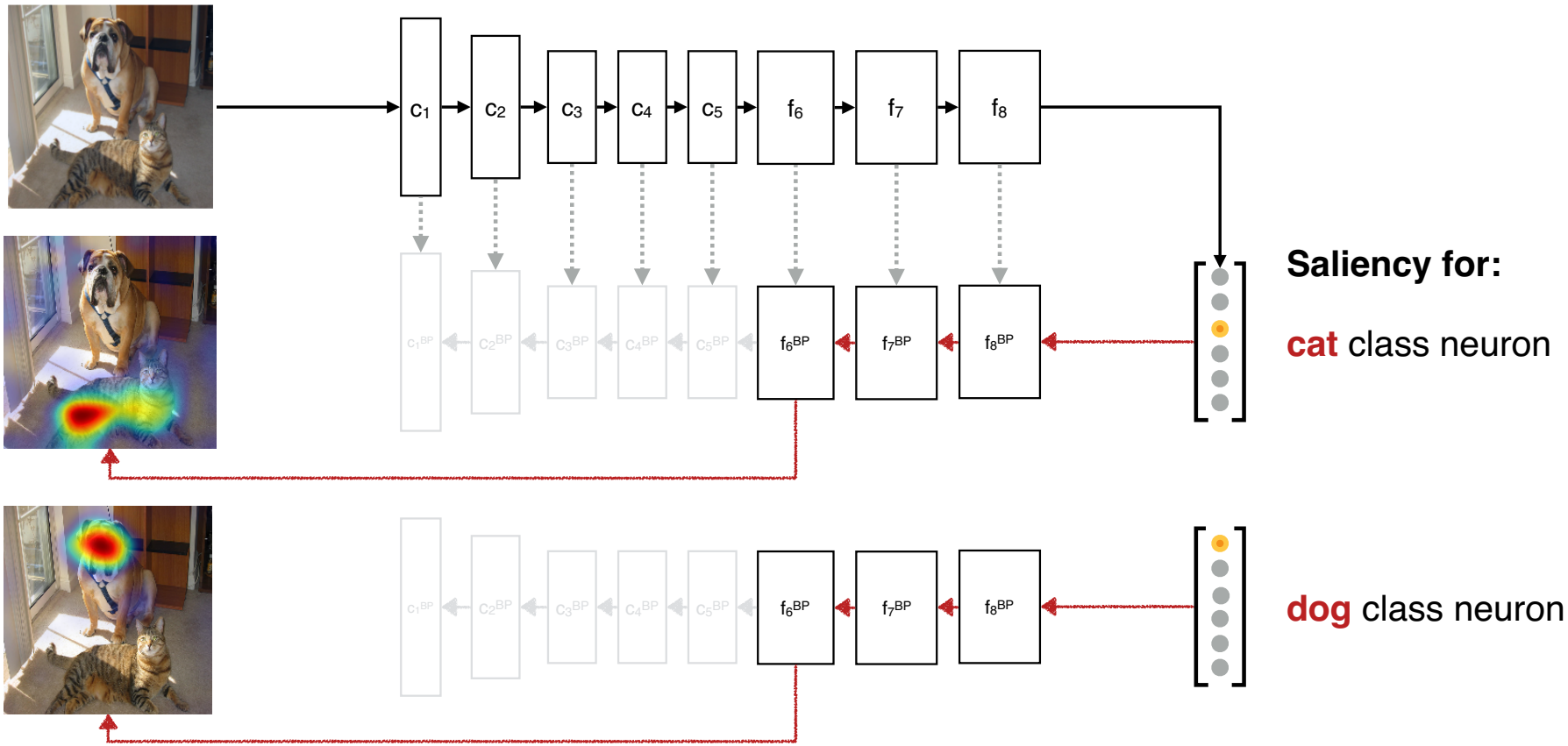
maximally activated neuron

random neuron

minimally activated neuron

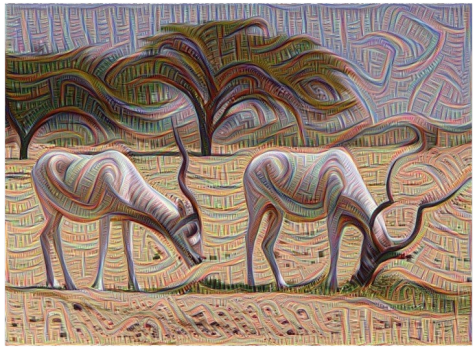
Cam and Grad-Cam

- Better channel specificity can be achieved by backpropagating only a few layers



DeepDream / Inceptionism

- Attempt to understand the inner workings of the network
- Optimize with respect to image
- Idea
  - Arbitrary image or noise as input
  - Instead of adjusting network parameters, tweak image towards high “X” where “X” can be
    - Neuron/Activation map/Layer
    - Logits/Class probability
  - Search for images that are “interesting”
  - Different layers enhance different features



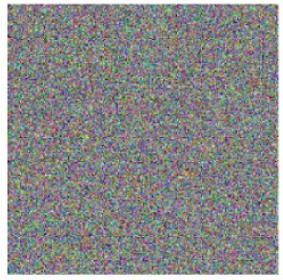
https://ai.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html

DeepDream / Inceptionism

- Find an image x such that the activation $\phi_n(x)$ at layer n is high

$$\max_x \phi_n(x) - \boxed{\lambda \mathcal{R}(x)}$$

Regularizer: e.g. L1 or L2 norm of image



- Algorithm
  - Forward propagate to layer n
  - No minimization of loss. Instead maximize L2 norm of activations of a particular NN layer
  - Backpropagate to input layer
- Resulting image will show learned features

https://ai.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html

Inversion

- Inversion attempts to construct an image from a layer activation $\hat{y}$:

$$\hat{x} = \min_{x}(\|\phi(x) - \hat{y}\|_2^2 + \lambda\mathcal{R}(x))$$

  - $\hat{x}$ is the reconstructed image
  - $\phi(x)$ is the network output for input image x
  - $\hat{y}$ is the desired activation
  - $\mathcal{R}$ is the regularizer

Imperial College London

Imperial College London

Robustness: Adversarial Methods

Daniel Rueckert
Department of Computing
Imperial College London, UK

Trustworthy AI/ML

Seven key requirements for trustworthy AI/ML

https://ec.europa.eu/digital-single-market/en/news/ethics-guidelines-trustworthy-ai

Adversarial attacks

- Unintuitive behaviour of NNs

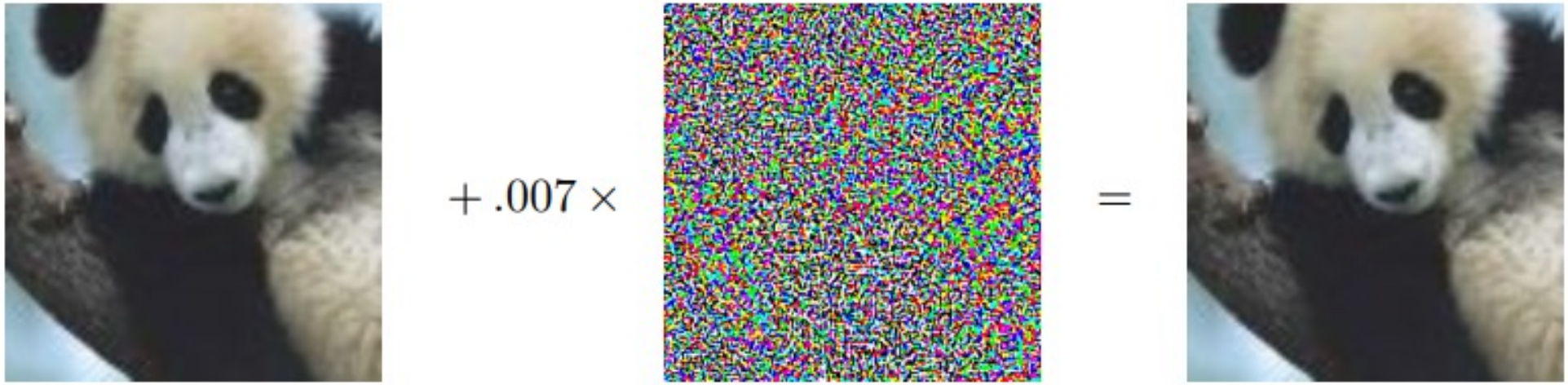
“panda”
57.7% confidence

“gibbon”
99.3 % confidence

I. Goodfellow et al. ICLR 2015

Adversarial attacks

- Unintuitive behaviour of NNs



I. Goodfellow et al. ICLR 2015

Adversarial attacks - Perturbation

- Assume a linear classifier:

$$\theta^T x$$

- We can think of an adversarial example that contains a small, non-perceivable perturbation to the input. Let's denote the perturbation as η:

$$\tilde{x} = x + \eta$$

- Then, the logits of the classifier would be

$$\theta^T \tilde{x} = \theta^T (x + \eta)$$
$$= \theta^T x + \theta^T \eta$$

Adversarial attacks - Perturbation

- Given a small perturbation η, the effect of the perturbation on the logits of the classifier is given by $\theta^T \eta$.
- Idea:
  - Find a η that causes a change that is non-perceivable and ostensibly innocuous to the human eye, yet destructive and adverse enough for the classifier to the extent that its predictions are no longer accurate.
  - An adversarial example is one that which maximizes the value of $\theta^T \eta$ to sway the model into making a wrong prediction

Adversarial attacks - Perturbation

- Problem:
  – Need a constraint on η; otherwise, one could just apply a large perturbation to the input
- Solution:
  – Apply a constraint such that

$$\|\eta\|_\infty \leq \epsilon \qquad \|\boldsymbol{x}\|_\infty = \max_i \{|x_i|\}$$

- Assume a perturbation:

$$\eta = \epsilon \cdot \text{sign}(\theta)$$

- What are the bounds of this perturbation?

I. Goodfellow et al. ICLR 2015

Adversarial attacks - Perturbation

- What are the bounds of this perturbation?

$$\eta = \epsilon \cdot \text{sign}(\theta)$$

$$\theta^T \eta = \epsilon \cdot \theta^T \text{ sign}(\theta)$$

$$= \epsilon \|\theta\|_1$$

$$= \epsilon m n$$

Here the average magnitude of an element of θ is given by m

I. Goodfellow et al. ICLR 2015

Adversarial attacks - Perturbation

- This means that the change in activation given by the perturbation increases linearly with respect to n (or the dimensionality).

- If n is large, one can expect even a small perturbation capped at ϵ to produce a perturbation big enough to render the model susceptible to an adversarial attack.

- Remember that for images $n = \text{no. of pixels}$

- Such perturbed examples as referred to as ***adversarial examples***

I. Goodfellow et al. ICLR 2015

Adversarial attacks - Fast Gradient Sign Method

Key idea:

- Perform gradient descent in order to maximize the loss (the goal of adversarial attack).
- Consider the input image x to be a trainable parameter and compute the gradient with respect to the input image to create a perturbation.

$$\eta = \epsilon \cdot \text{sign}(\nabla_x \mathcal{L}(\theta, x, y))$$

- An adversarial example can be created as:

$$\tilde{x} = x + \epsilon \cdot \text{sign}(\nabla_x \mathcal{L}(\theta, x, y))$$

I. Goodfellow et al. ICLR 2015

Adversarial attacks

How can you use adversarial attacks?

1. Generate adversarial examples
2. Add the generated adversarial examples to the training set
3. Retrain model using training set

Adversarial data augmentation

References

- Trustworthy ML
  - https://ec.europa.eu/digital-single-market/en/news/ethics-guidelines-trustworthy-ai
- Federated Learning: Challenges, Methods, and Future Directions
  - https://arxiv.org/pdf/1908.07873.pdf
- Privacy-Preserving Deep Learning
  - https://www.cs.cornell.edu/~shmat/shmat_ccs15.pdf
- Communication-Efficient Learning of Deep Networks from Decentralized Data
  - http://proceedings.mlr.press/v54/mcmahan17a/mcmahan17a.pdf

References

- The Future of Digital Health with Federated Learning
  - https://arxiv.org/abs/2003.08119
- Secure, privacy-preserving and federated machine learning in medical imaging
  - https://www.nature.com/articles/s42256-020-0186-1

References

- Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV)
  – https://arxiv.org/abs/1711.11279
- The building blocks of interpretability
  – https://distill.pub/2018/building-blocks/
- Understanding deep neural networks through deep visualization
  – https://arxiv.org/abs/1506.06579
- Visualizing and Understanding Convolutional Networks
  – https://arxiv.org/abs/1311.2901
- Understanding Neural Networks Through Deep Visualization
  – https://arxiv.org/abs/1506.06579