Department of Computing

Imperial College of Science, Technology and Medicine

Causality and Generative Models

This week's paper is [1]

Author: Anton Zhitomirsky

Contents

1 Trustworthy AI/ML

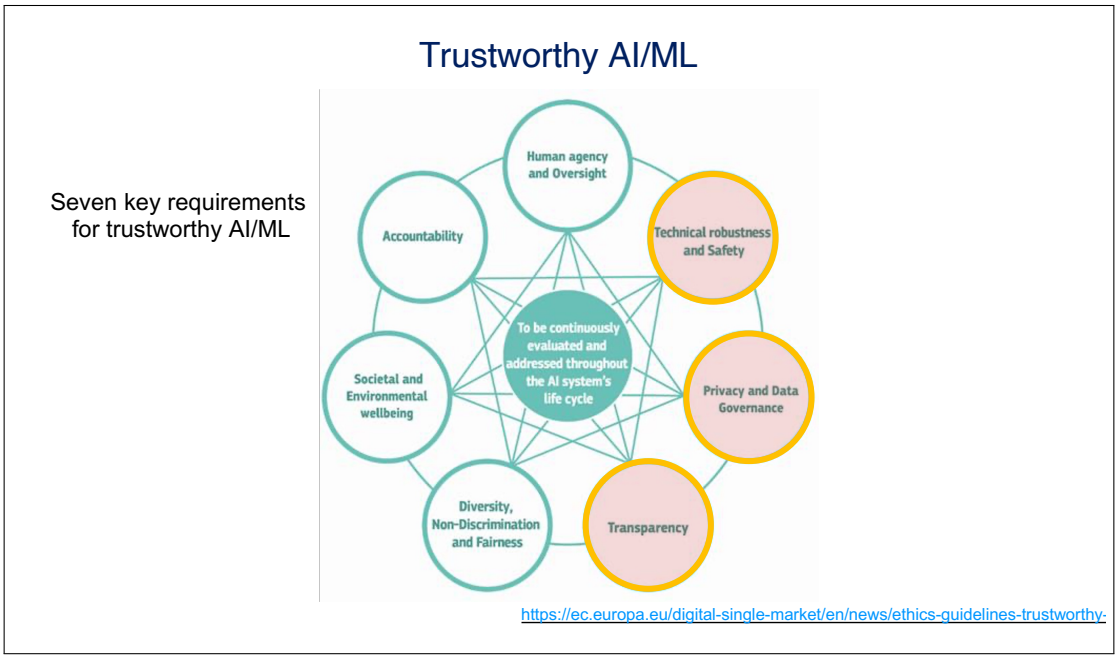


Figure 1: Some of these concepts don't relate to technical aspects, but general. This is the definition of the EU of what makes AI trustworthy and Reliable.

1.1 The need for data

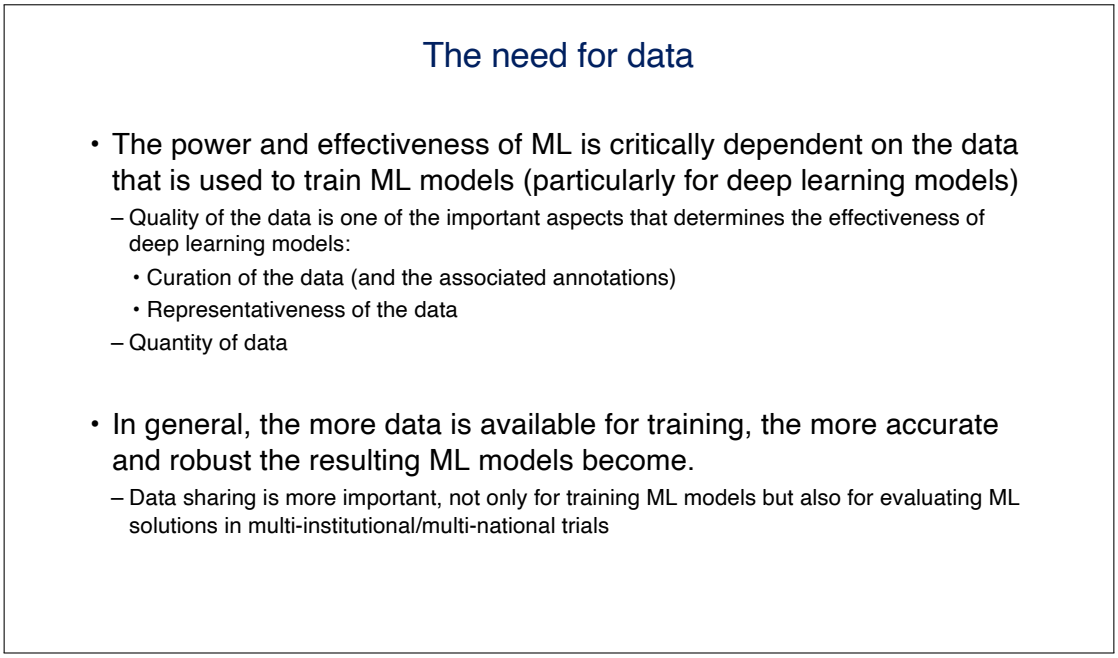


Figure 2: In order to build systems which wordk well we would like to use as much data as possible to train these models. However, the quality of the data also needs to be good (requires diversity, representative of problem and applies to the annotations).

1.2 What are the hurdles to getting more data?

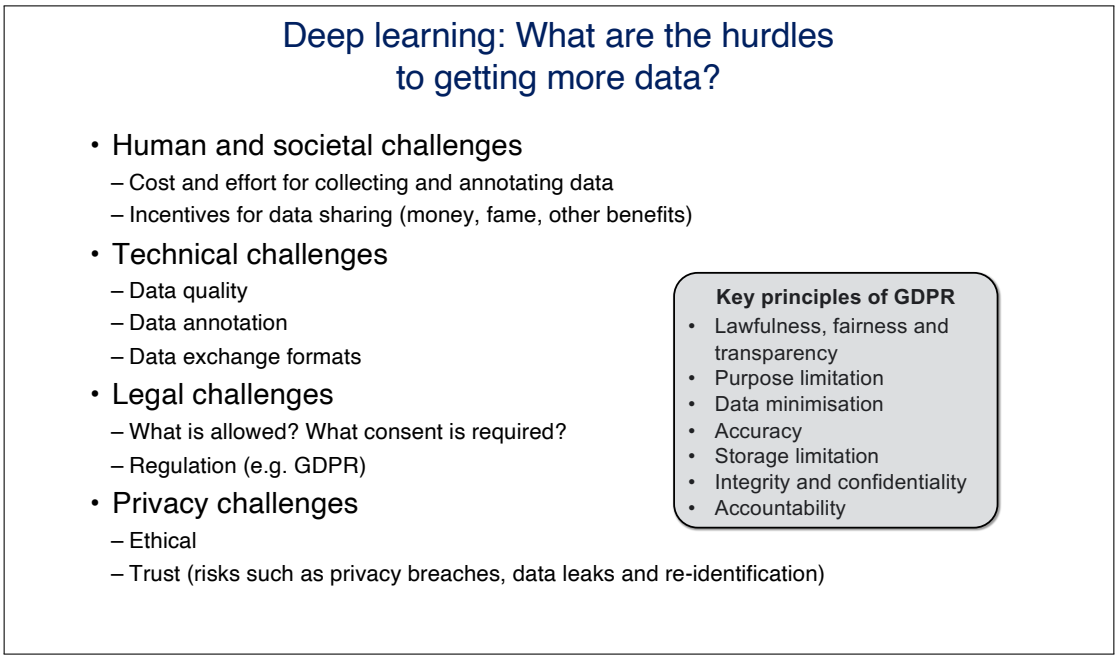


Figure 3: For different quality of data and annotation, we can weight these differently in the loss function.

1.3 Secure and Privacy-aware ML

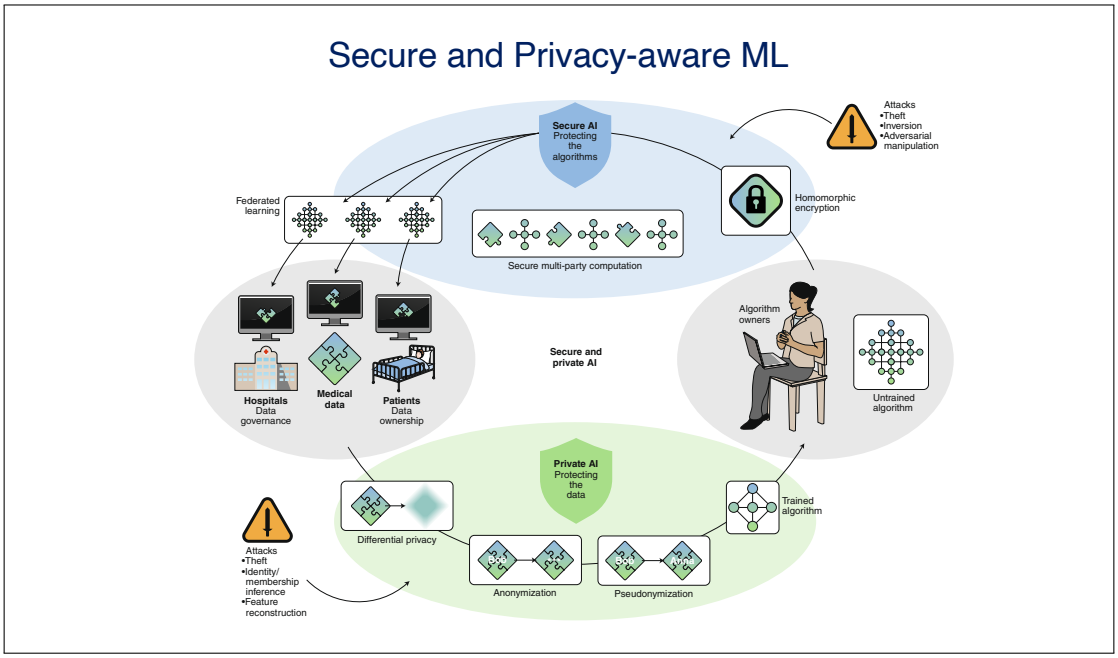


Figure 4: The concepts here apply to all imaging problems. Federated learning controls the flow of data. All of these techniques attempt to protect the computation.

1.4 Secure and Privacy-preserving ML

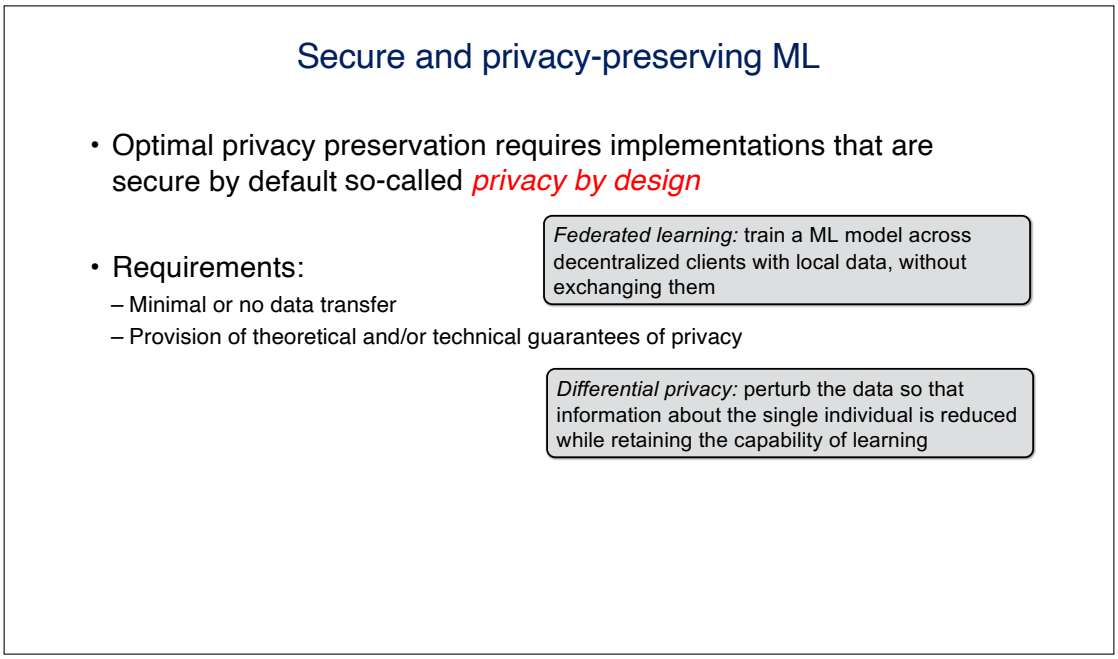


Figure 5: We can't retrospectively take an approach and hope you can build techniques which make sure that you ensure privacy. You need to design it form first principles and there are two overriding principles for this privacy by design. These are above. For minimal data transfer you can use federated learning, this uses decentralized learning with a bunch of clients with their data locally. The data is never transferred to the server, but only the result. Here, federated learning doesn't really guarantee privacy, but it gives the user control of their data. The second part, is differential privacy which anonymised the data.

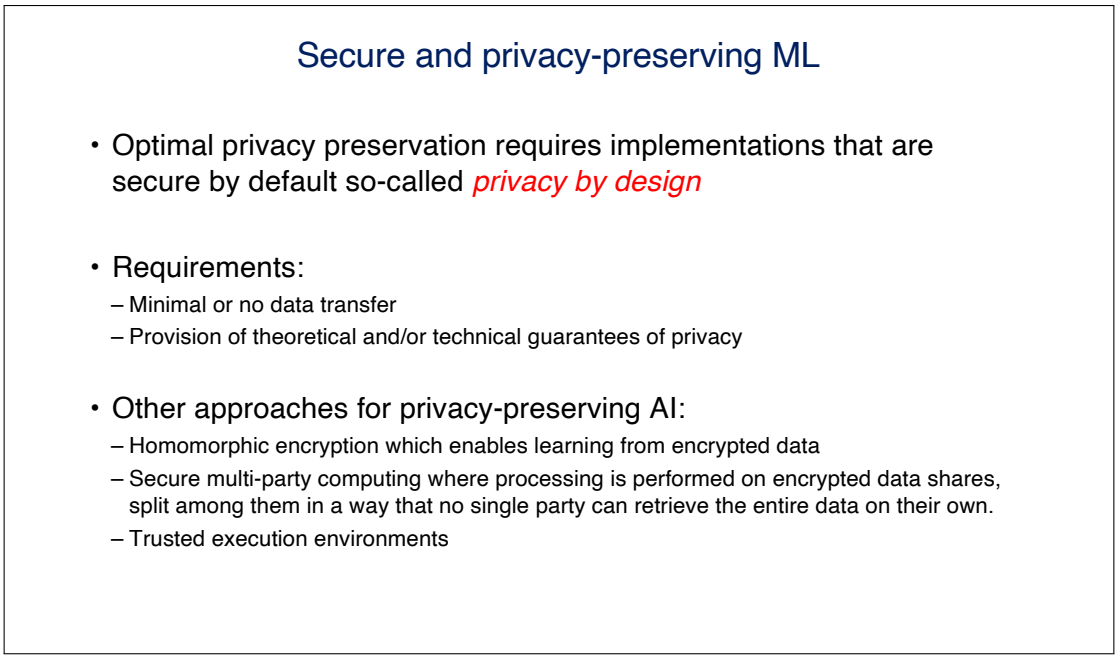


Figure 6: There are other add-ons you can use. The techniques are listed above.

1.5 Federated learning

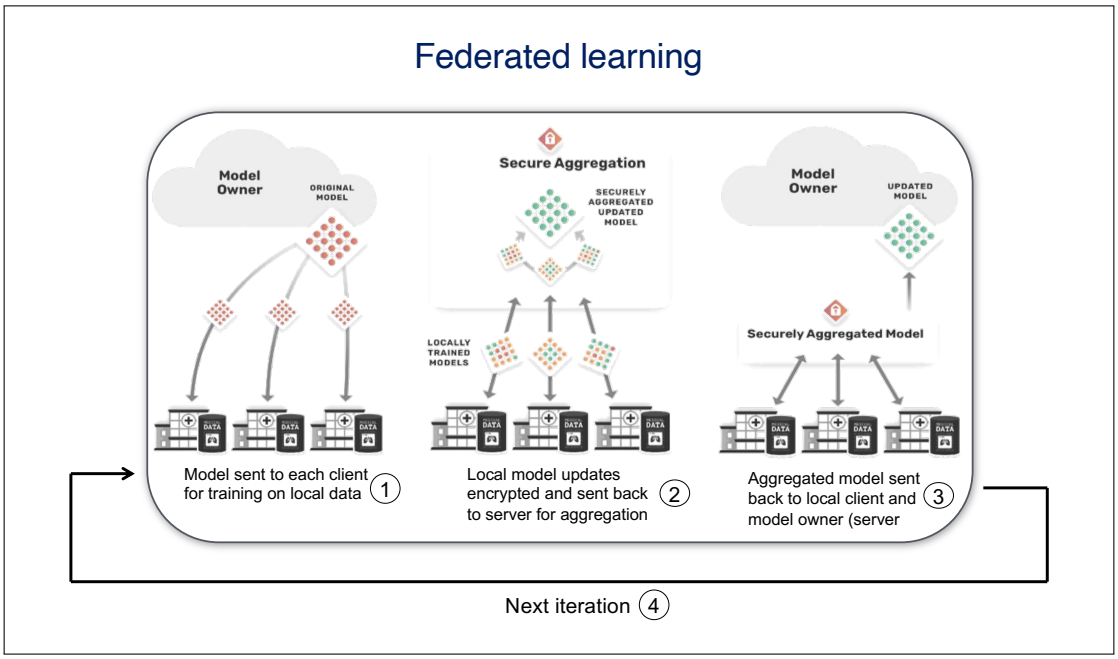


Figure 7: I have a central server where each client sends their data to a central cite. The data is sensitive, and the data owners are not allowed by law to share their data. Federated learning makes you send your ML model to each of the cite and each cite update the model parameters and sends it back to the central cite. The central cite does some aggreagation and you repeat the process. Importantly, is that the data stays with the owners.

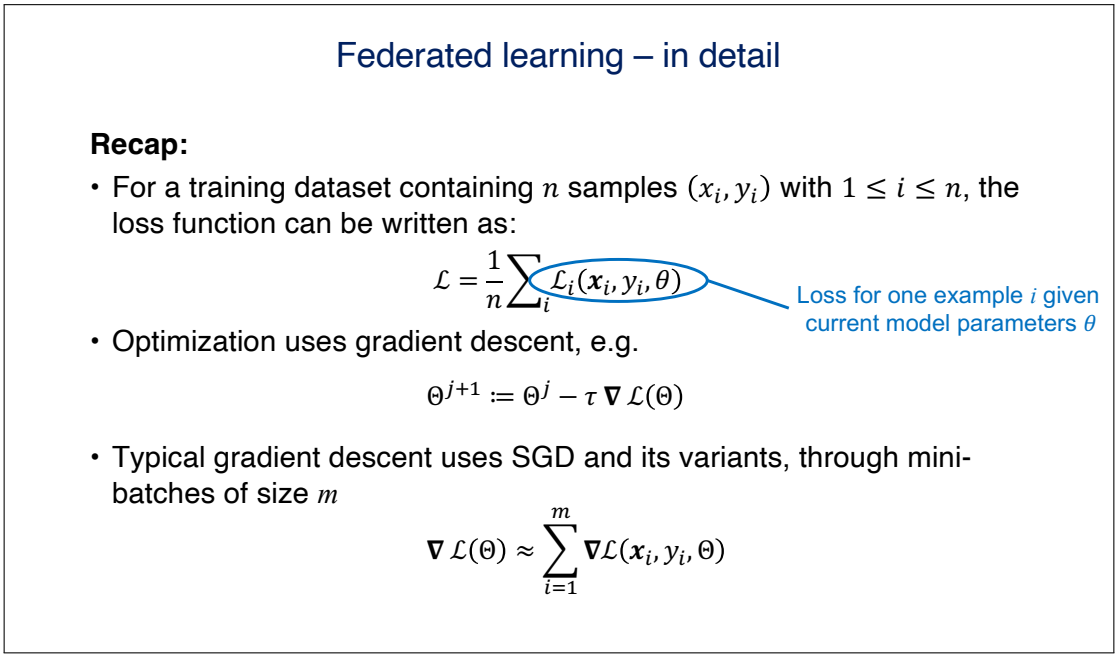


Figure 8: This is the standard learning process. If using stochastic gradient descent you aprpoximate your gradient by just looping over all the samples in the minibatch.

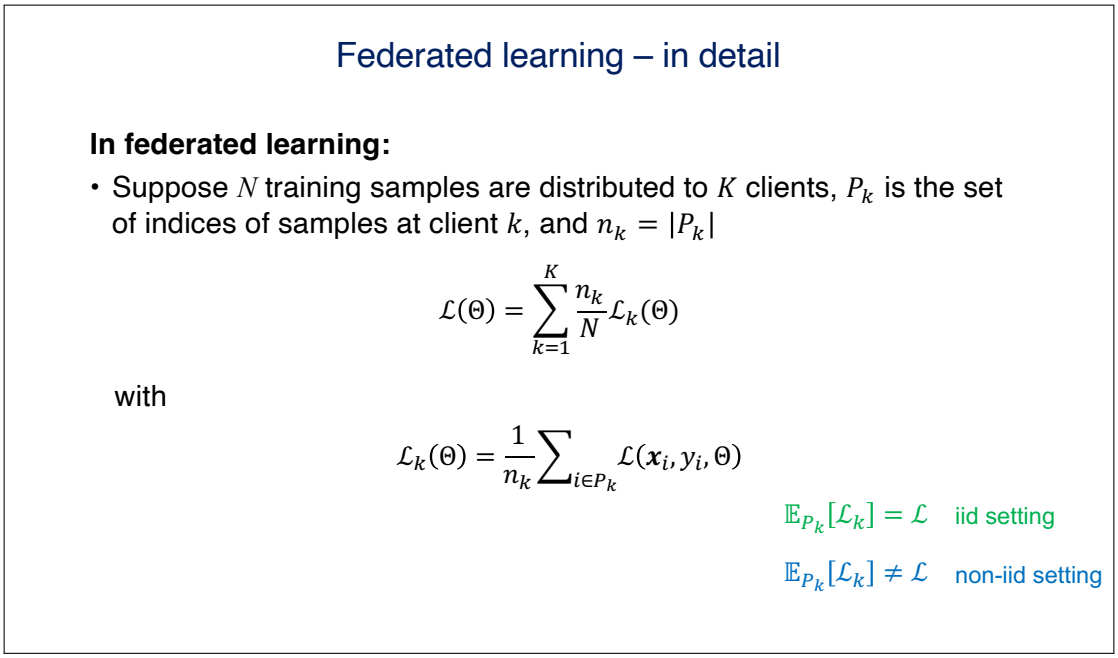


Figure 9: If you do federated learning, you assume that the training samples are distributed to K clients. you can still compute the loss function by computing the loss function from each cite buy weighting it depending on how many or how big the training set is. The first eqution is the standard loss PER CLIENT. If you sum this, you get the standard loss function.
If you compute the loss this way, the expectation of this loss computed this way for each client is equal to the overall loss if you assume the iid setting. If you have huge amounts of data in homogeneity lets say (a hospital in US may have different population characteristics to hospitals in EU), then each individaul loss terms may not be in the expectation equivalent to the overall loss. This impacts how you train.

1.5.1 Federated SGD

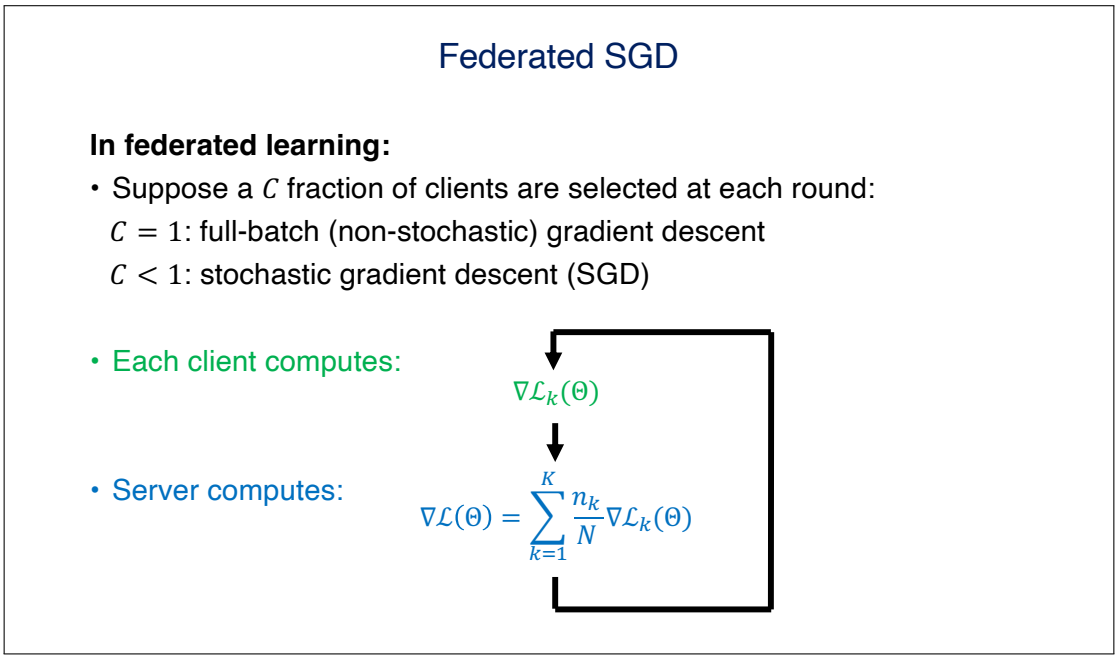


Figure 10: Assume that each client independently computes the loss and derivative for their dataset. What you transmit is the gradient

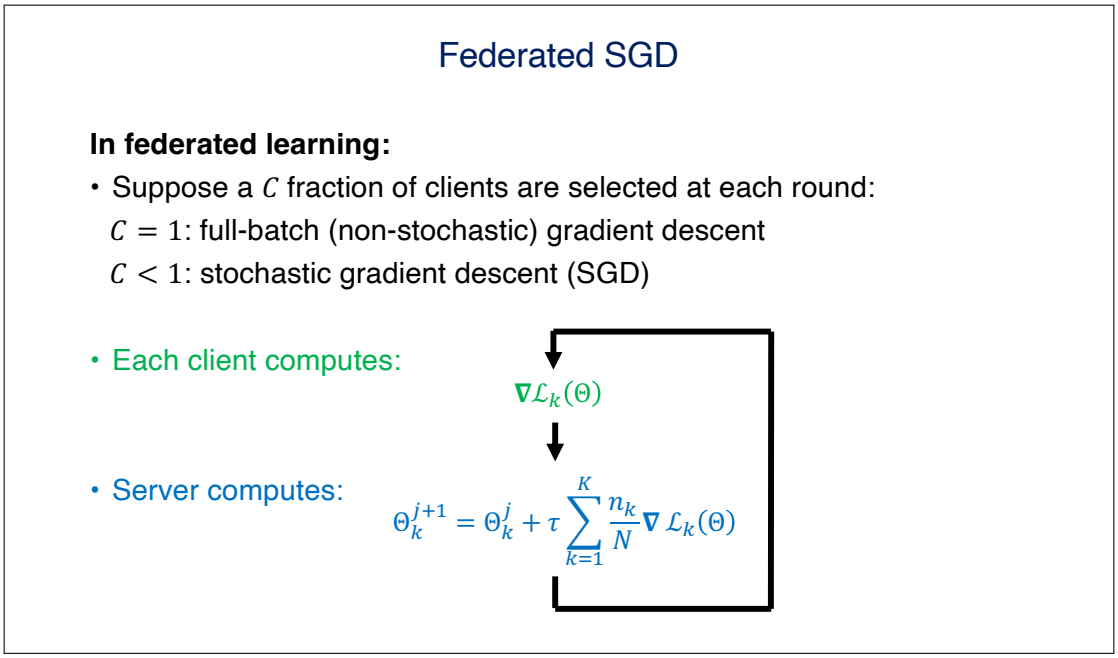


Figure 11: Then the server updates the model after combining all gradients.

1.5.2 Federated Averaging

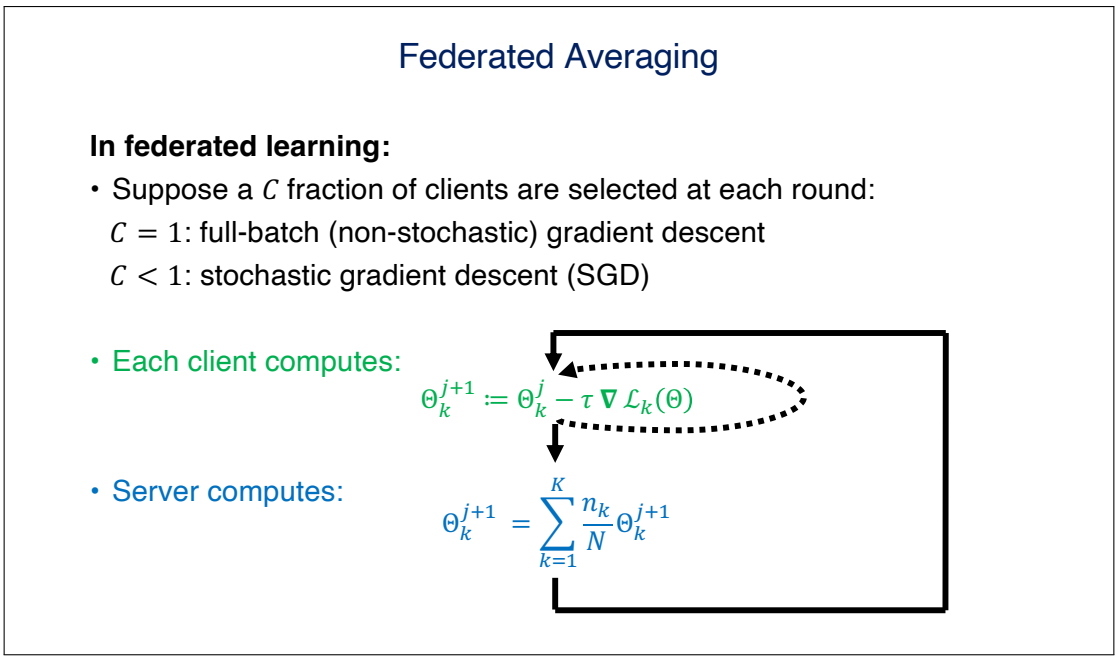


Figure 12: There is an alternative approach; the client transfers the gradient to the server, but you can also locally update the parameters by performing on each client a gradient descent separately. Then generating a new set of parameters and iterating a few times. Then sending out to the server back an updated set of parameters. On the server side, you then update the parameters by computing a weighted average of the parameters

1.5.3 Differences

The difference between the two approaches is that each client may not have that much compute power, and each client may typically have only one dataset. It doesn't make sense to do gradient descent on my local set becuase I have a very noisy gradient. This is becuase I'm limited to a batch size of one. In

the first appraoch, you send the gradient, but you accumulate the gradient and perform much more reliable gradient descnet. However there, you require constant communication between the client and the server. In the second example, you force each client to do 10 iterationso, and only then they share updated parameters. There, you have better control over the communication between client and server.

1.5.4 Algorithm

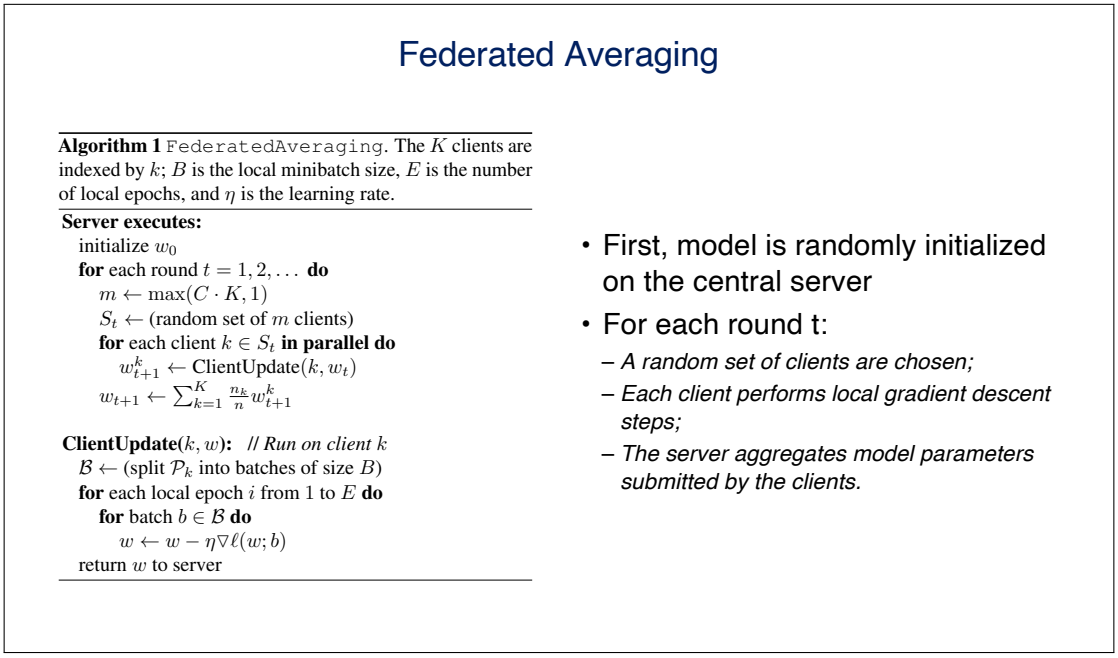


Figure 13: caption

1.5.5 Challenges

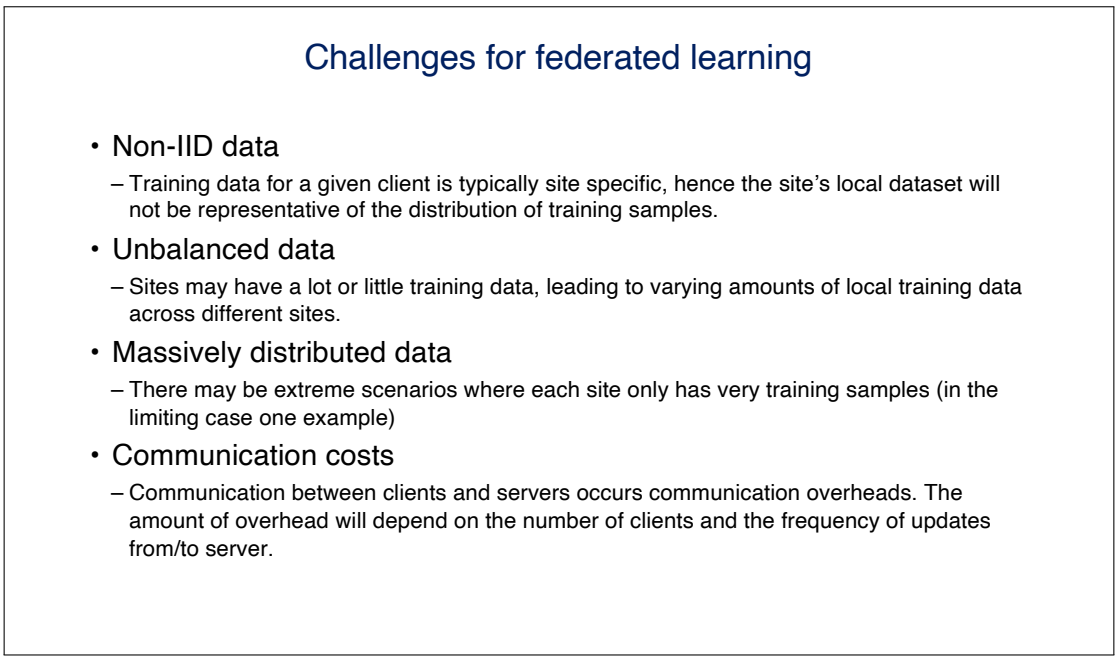


Figure 14: 1) approximation of whole gradeints no-longer approximates the overall gradent. 2) updates might not be as reliable on some cites as others 3) this is extreme and inefficient.

1.6 Homomorphic Encryption

There is still an extent of sensitive data being transferred.

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

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

- Here:

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

Figure 15: finds an encryption scheme, such that the encrypted data can be manipulated in the encrypted domain as though it had been decrypted.

1.6.1 ML in clinet/server setting

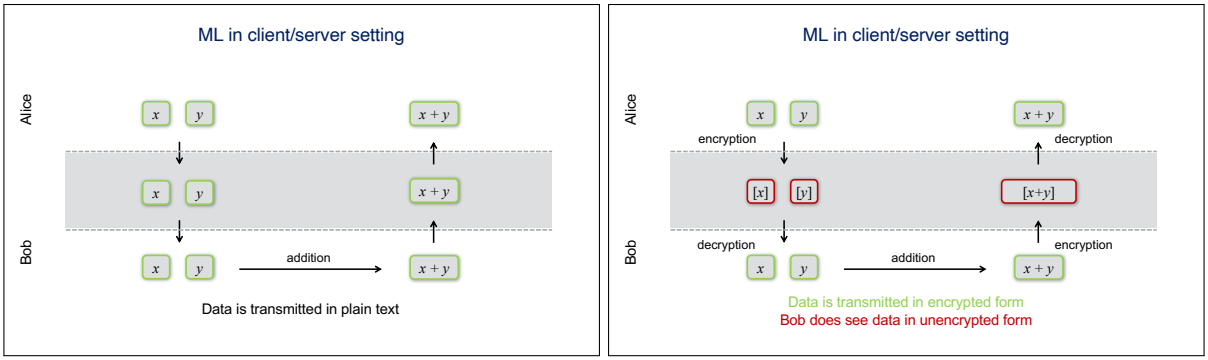


Figure 16: If Alice is the client and Bob is the server, then sending the data to the server, performing the addition, and sending it back is insecure. With standard encryption, we protect it through encryption, then we decrypt it because we need to decrypt it.

1.6.2 Homomorphic Encrypion

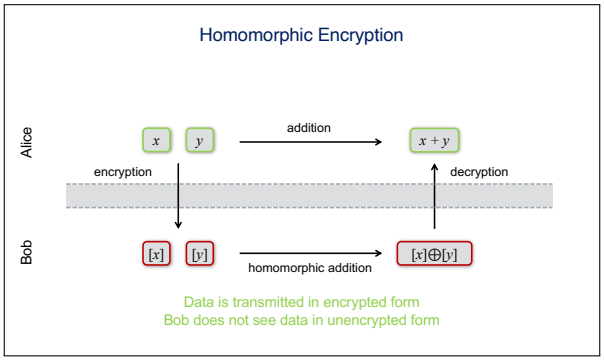
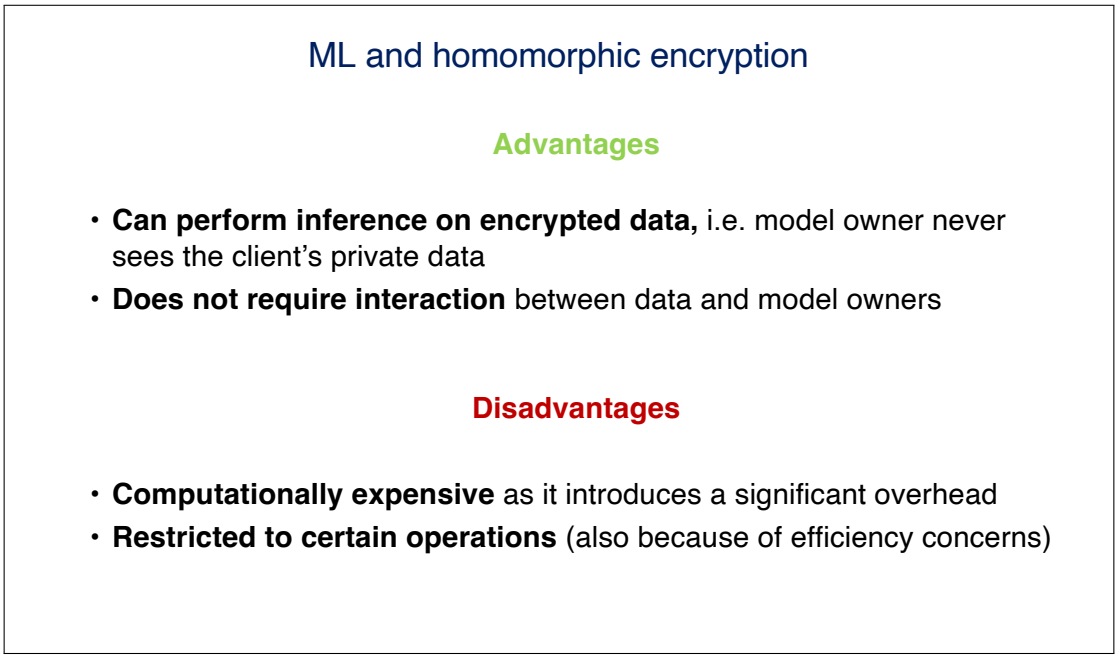
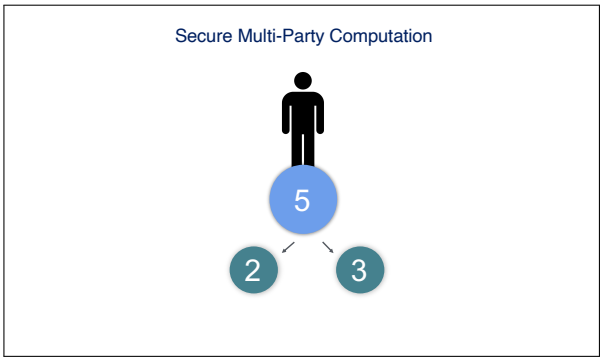


Figure 17: In homomorphic encryption, we can perform the addition in the encrypted domain. Bob (the server) never gets to decrypt the data. However, this is computationally expensive (100x slower for deep learning and also some mathematical operations are hard to find a homomorphic encryption for).

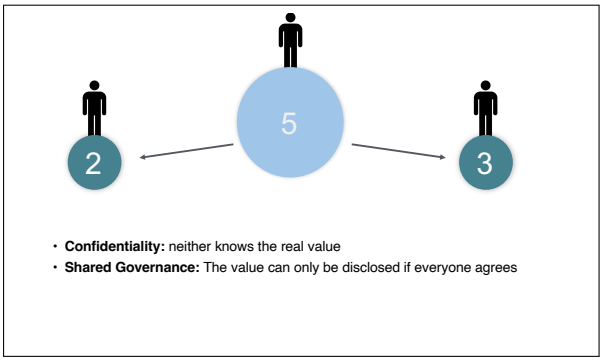
1.6.3 Advantages and Disadvantages in ML



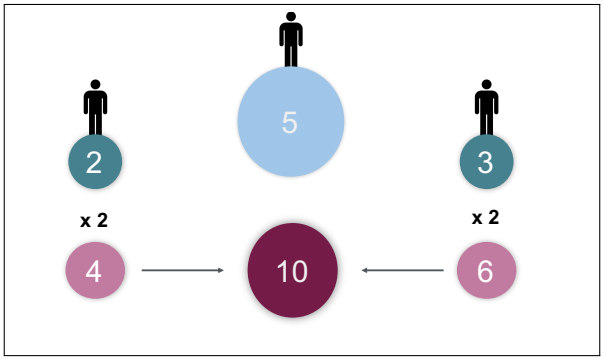
1.7 Secure Multi-party Computation



(a) There is an alternative. If you have a secret value of 5, you can split it into two parts (you don't tell anyone how you split it)

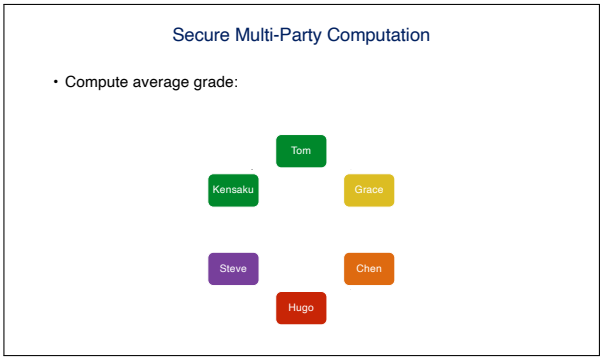


(b) We send it to clients

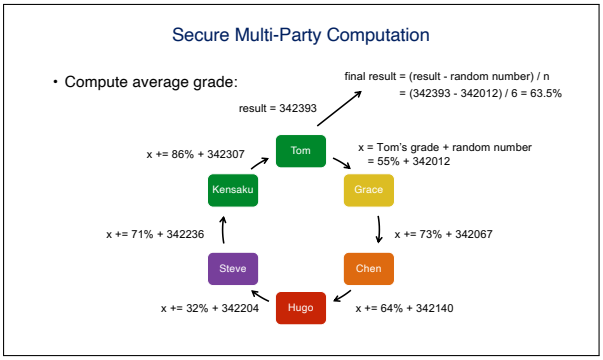


(c) the clients perform some computation on their share of the data and return the answer back to you.

Figure 18: You are then able to reconstruct the answer because you know how you split it. The client never saw the original value.



(a) Suppose everyone has exam grades but don't want to share them



(b) We can do a circular calculation by adding a random number to it (only the first person creates a random number). Then this process continues until we get back to the first person. There is no way to know individual exam grades because many random numbers have been added. Then, the first participant knows what the random offset was; they subtract it and find the average.

Figure 19

1.8 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.

Figure 20: In a medical dataset, a dataset is anonymized if it doesn't contain identifiable data. However, in reality it is not enough. Record linkage attacks are possible to reverse engineer.

1.8.1 k-anonymity

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.

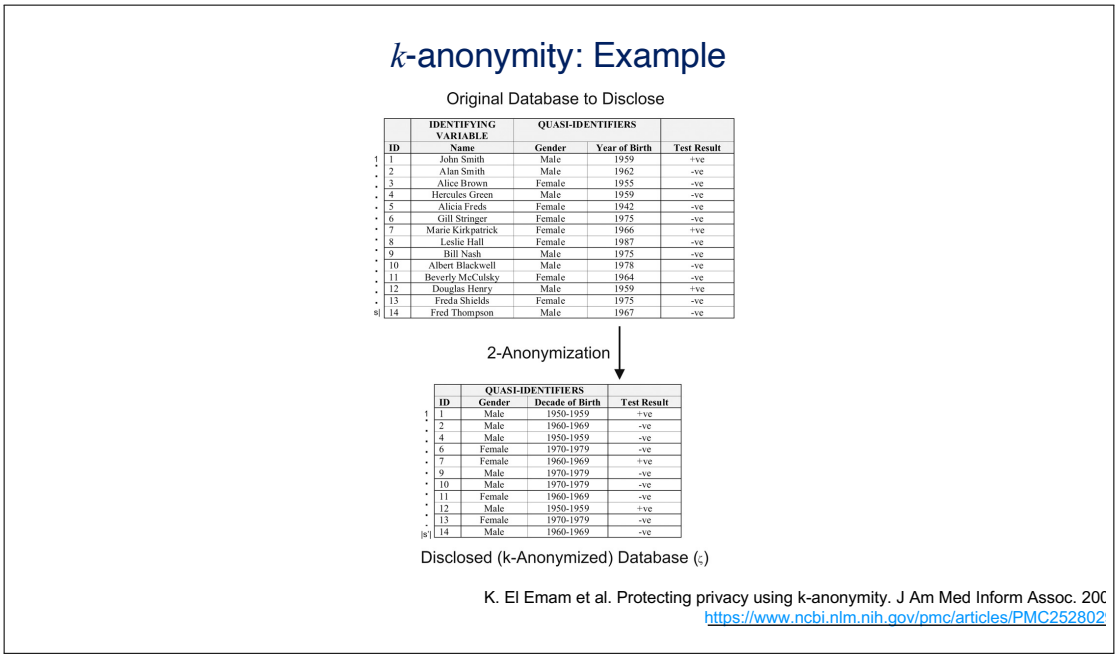


Figure 21: You can annonymize this dataset by removing the name, but the date of birth gives us a high chance of re=identifying someone.

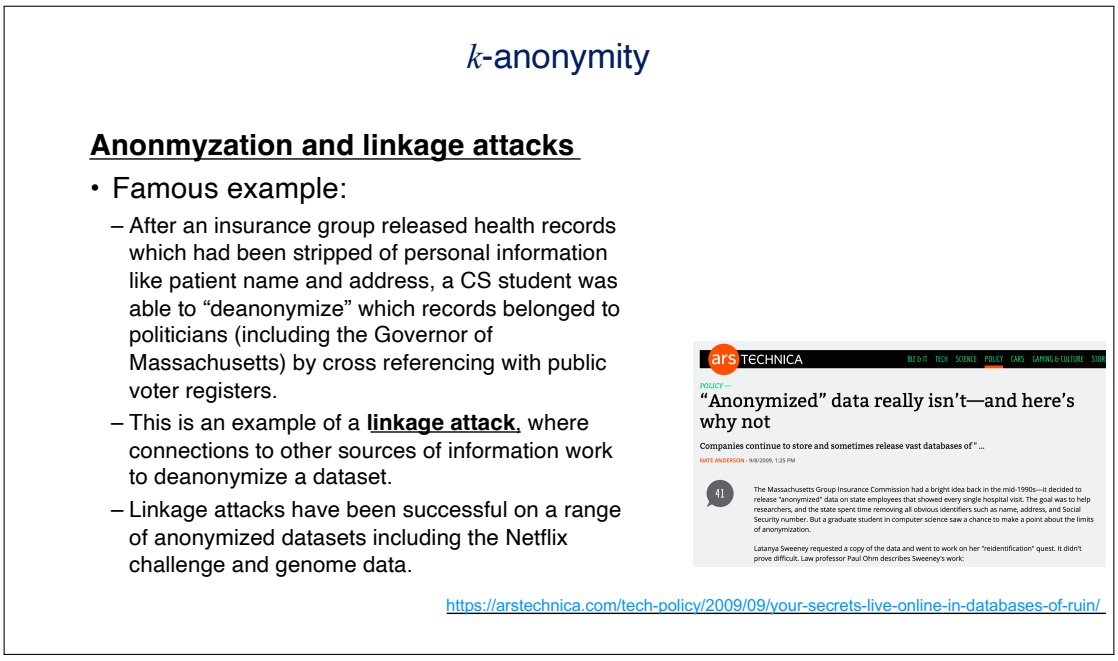


Figure 22: This doesn’t work in practice as it doesn’t make any assumptions about what side information you have. Here, we used additional informatino even in the public domain to re-identify.

- 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.

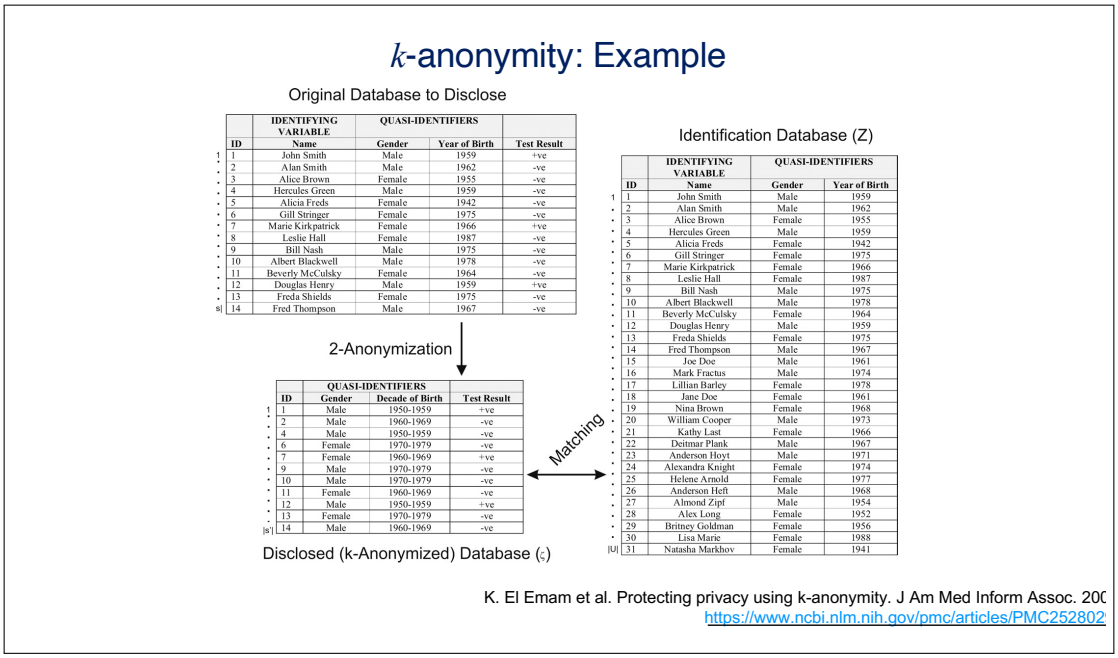


Figure 23: Here is an example; this relies on you having access to another dataset and match for example with this other dataset individual patients and to work out what their identity is.

1.9 Privacy attacks

1.9.1 Model inversion

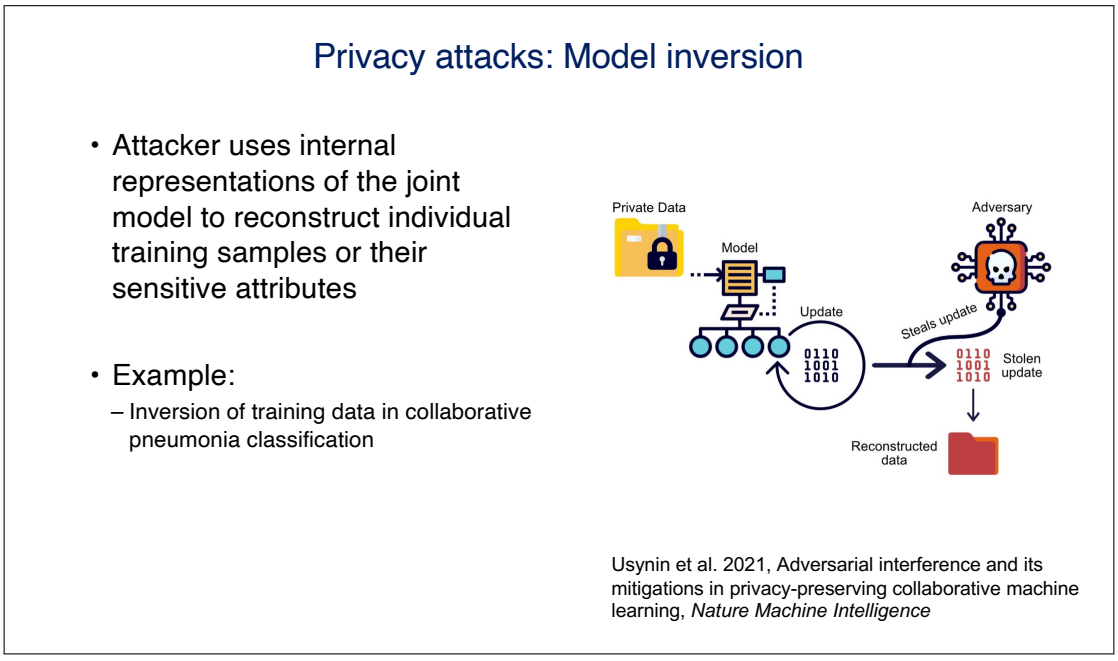


Figure 24: In the context of deeplearning, you worry most about 3 different pricay attacks you can do with images. The first attack is to try and invert the model to get the training samples.

1.9.2 Membership inference

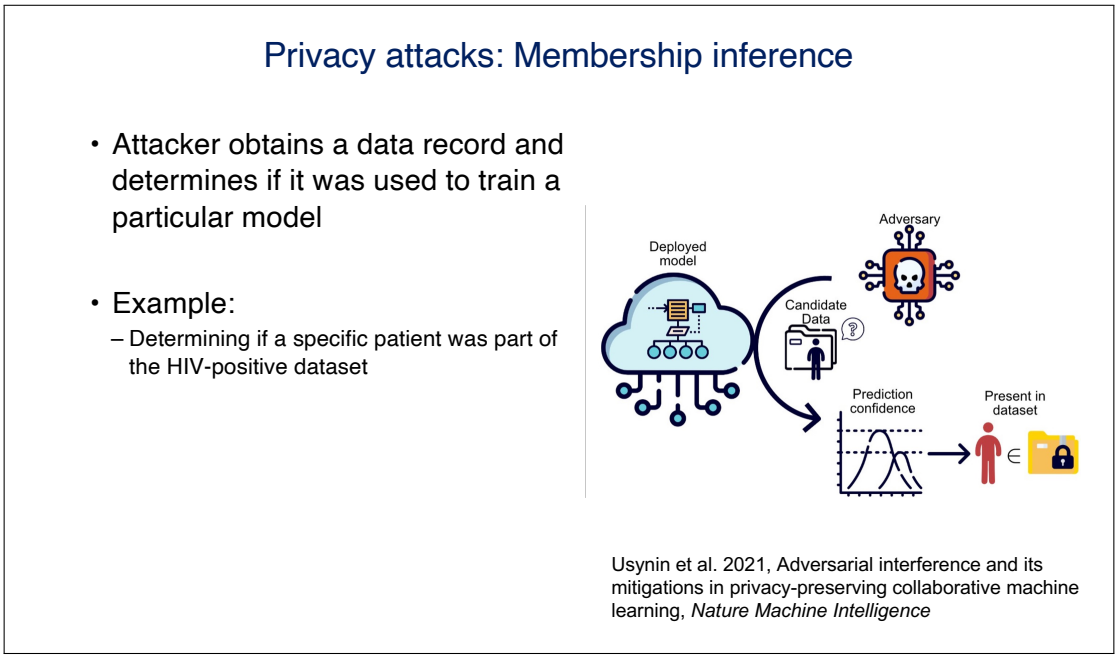


Figure 25: Second type tries to work out if a particular dataset is in was used in the training set.

1.9.3 Attribute inference

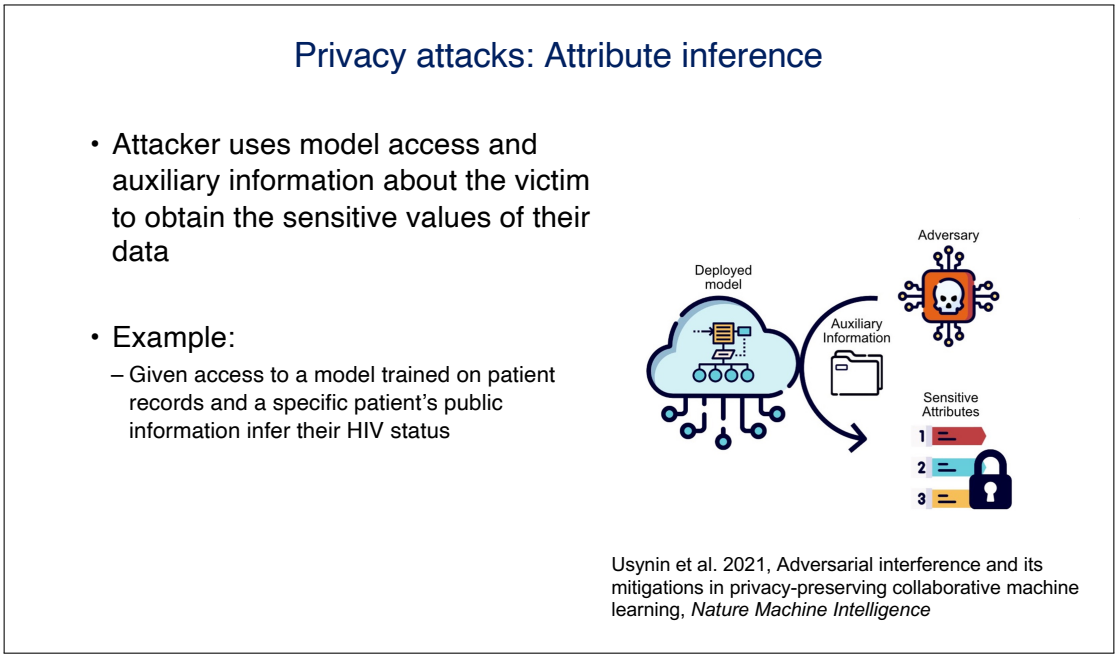


Figure 26: Finally, don't reconstruct entire data but only some sensitive data or informatino about the data. can be common problem.

1.10 Differential Privacy

These attacks are preventable

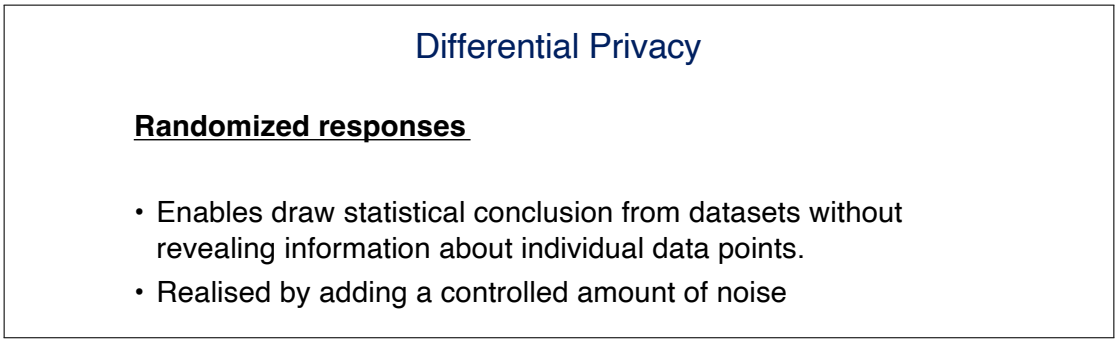


Figure 27: You randomize certain parts of the data without randomizing it enough where you destroy the data.

1.10.1 Randomized responses

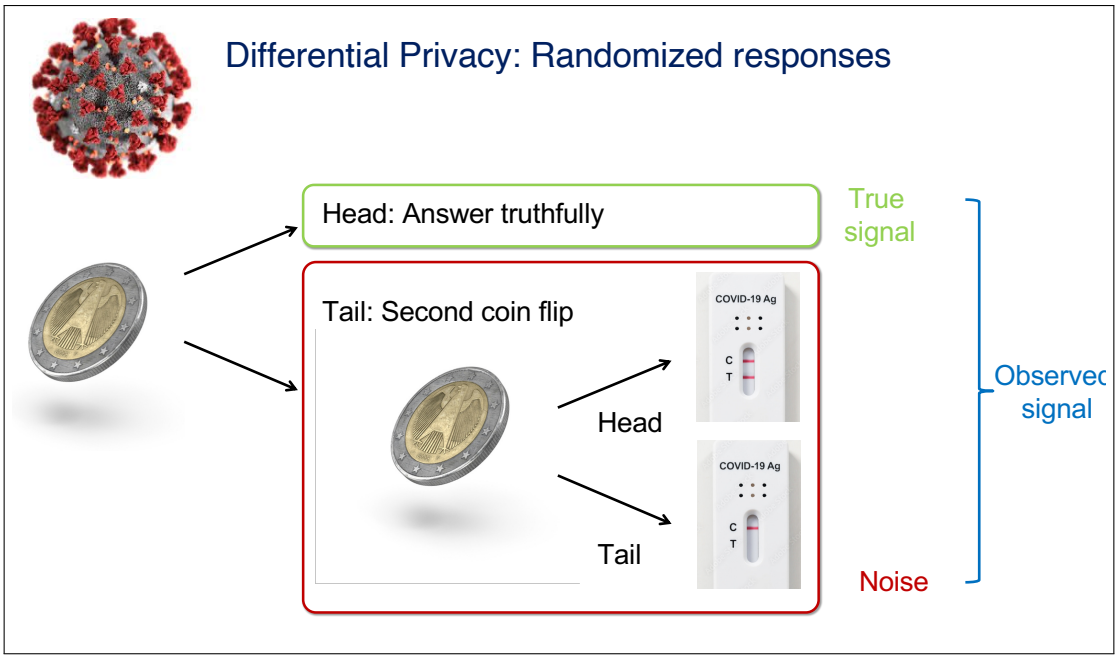


Figure 28: You want to find out who has coronavirus. But you don't really want to admit it. Still, you want some statistics. Therefore, each flip a coin and don't tell you the result of the coinflip. Following the flow above, you tell the answer. This produces a noisy answer for those that had tails. This plausible deniability means that you can't make any assessments on the individual level. But if you count the number of votes that are positive and negative, you can still calculate the prevelance.

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.

4

Figure 29: This can be applied to any machine learning model

1.10.2 Pracitcal Application of Differential prviacy

Differential privacy

- An algorithm can be made approximately invariant to inclusion of a single data sample
- This is achieved through the addition algorithm

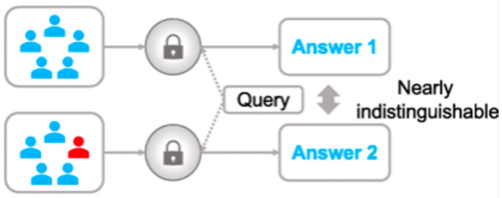


Figure 30: even if you leave one person out you want the model to give virtually the same ansewr. This is a good thing becuase you don't know which person was left out and you cant infer anything about the person. "This is nearly indisitinguishable. You have a paramerter ϵ which tells oyu how strict the privacy guarantees are. for strict privacy guarantess (small episilon) with small sample size this won't work, so you'll have to add so much noise that the output of the model utility will drop down to zero."

Differential privacy (continued)

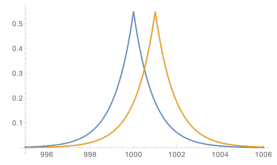
- This can be mathematically defined using the following (relaxed) expression:

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:



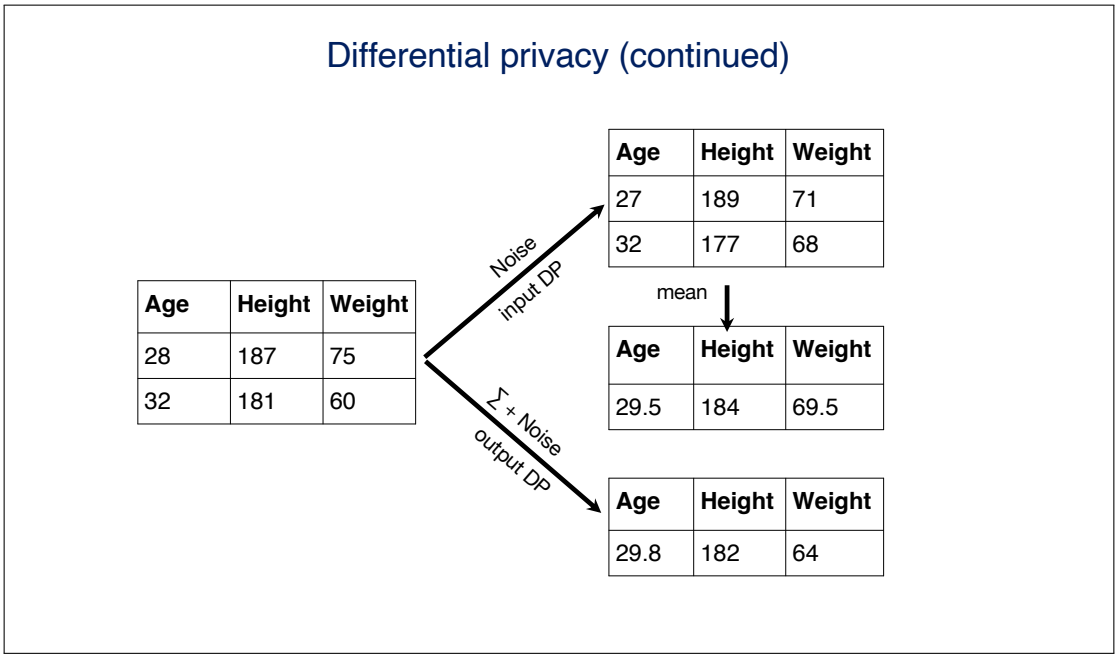


Figure 31: If you do this with tabular data you can add randomized noise to each entry. You can do this with input data or output data. With images, you do it with the input.

1.10.3 Concrete Mitigation: stochastic gradient descent (DP-SGD)

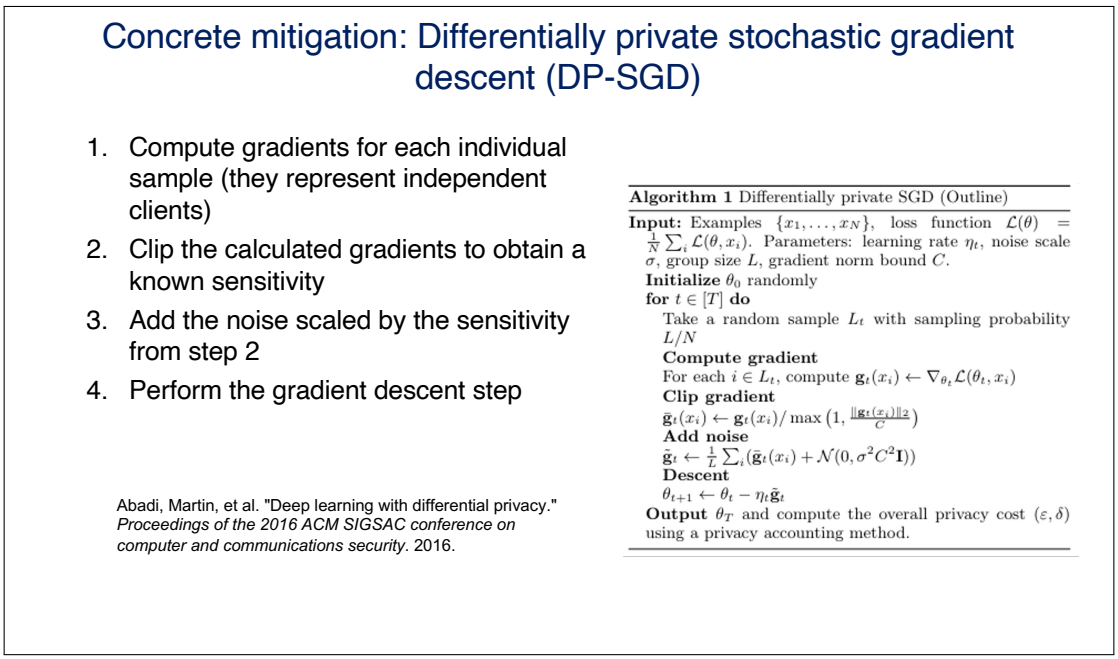


Figure 32: It performs one extra step where you compute the gradient where you clip your gradeint and add noise. Clipping limits the norm of each gradient; limiting the influence that each training example can have on your overall gradient. Since noise is noramlly distributed this should average itsself out. This completely destroyed the model's ability to reconstruct training data!

When you force deep learning to not force the model to memorize the data, you're instead normalizing and preventing overfitting, since often regualirsation includes adding noise.

2 Interpretability and Explainability

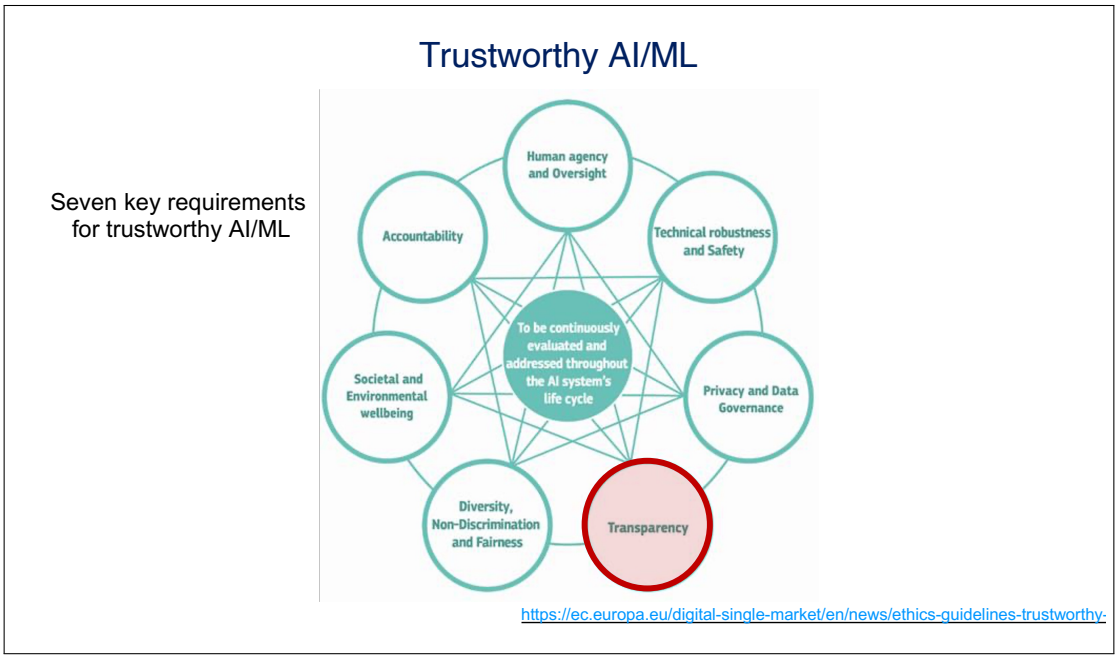


Figure 33: caption

2.1 Interpretability

2.1.1 Why is it important?

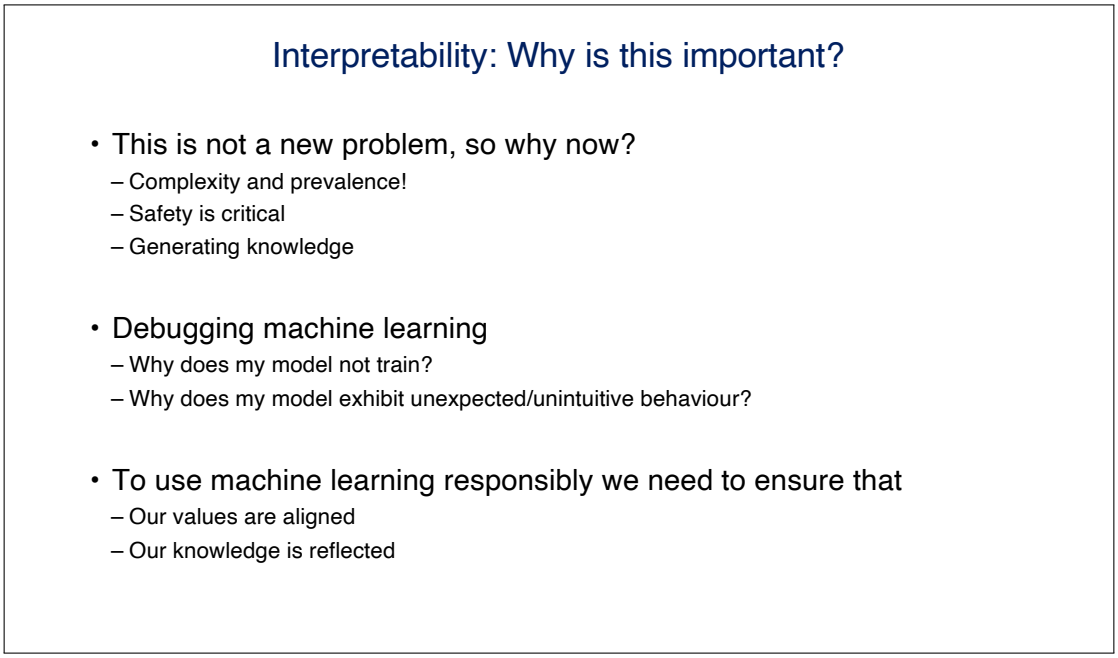
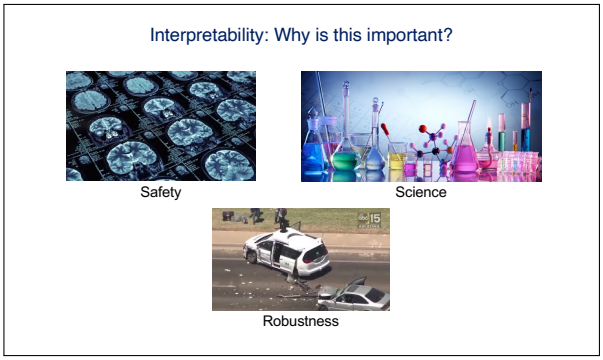
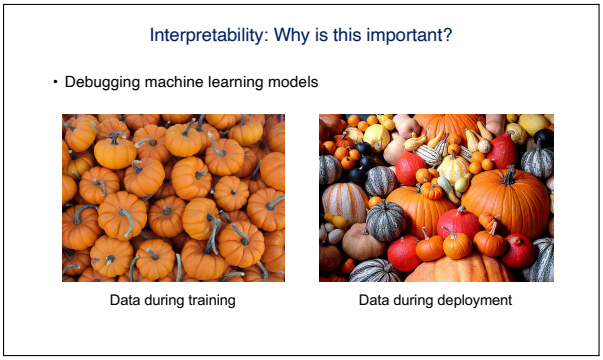
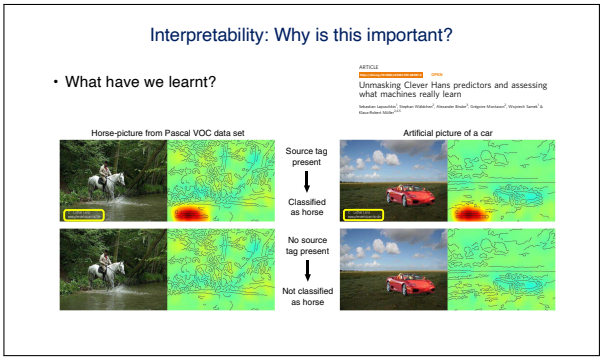


Figure 34: Useful to gain insights into what your machine learning does. Furthermore, it is useful to use as a debugging tool

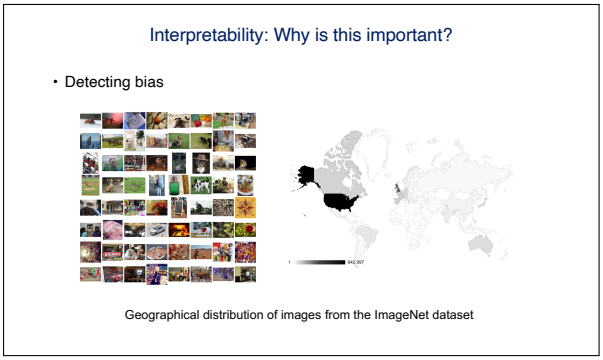




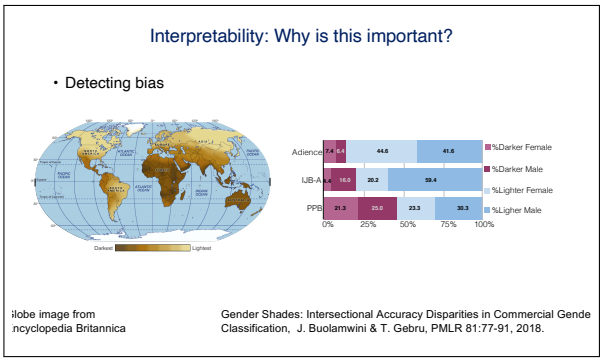
(b) a classifier might not work on the right, in some cases, it might not be becuase of the trianing data.



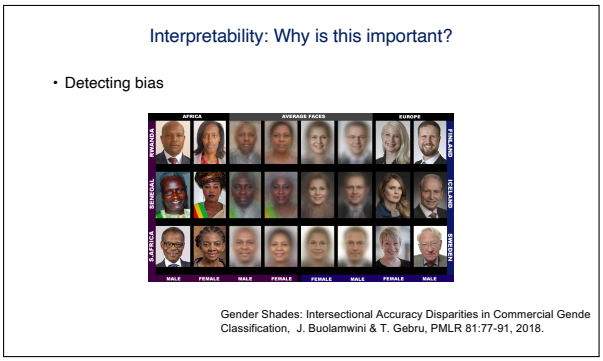
(c) Here, a model is trained to predict whether there is a horse or car in the picture. Some of the data in the dataset had text. This text tells you a lot about the image. It might not be obvious nor apparent, but an interpritable method allows you to query this model and extract which features in the image lead to the classification



(d) Data distribution amongst countries may not work in other countries



(e) This also is bad with face detection for skin colour



(f) It can be useful for detecting bias in the models

2.1.2 Common misunderstandings

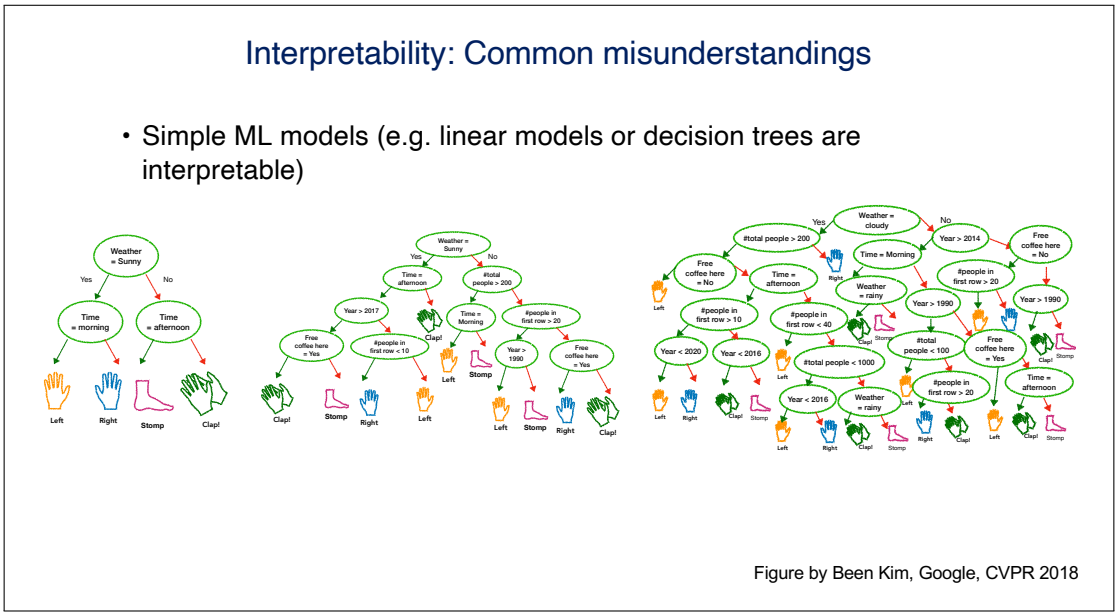


Figure 35: decision trees are interpritable, but these can get very complicated and no longer interpritable.

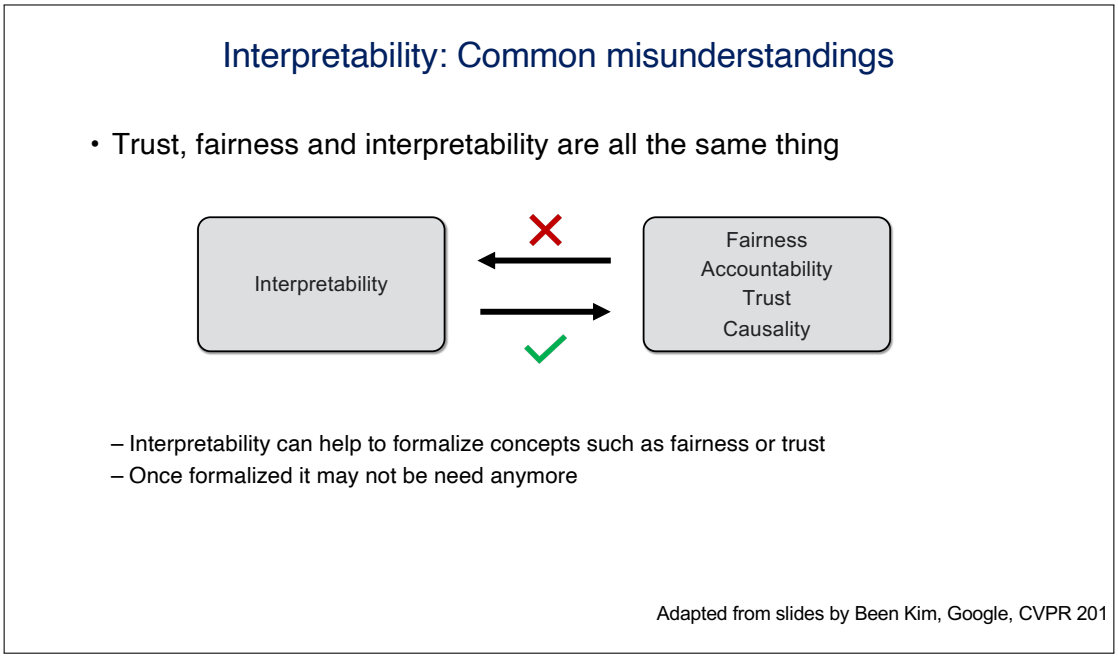
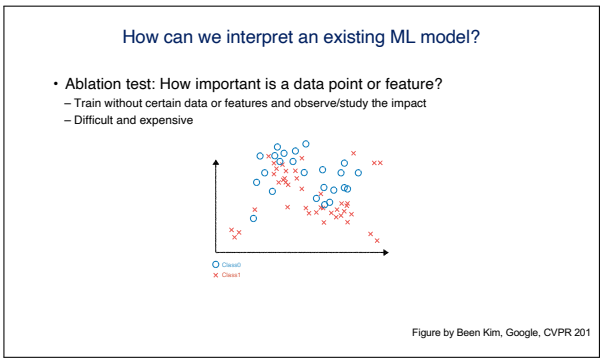


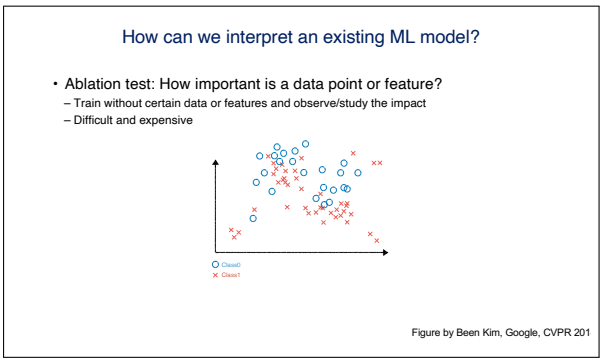
Figure 36: If you want to trust a method, you need it to be interpritable. But becuase a method is trustworthqy, it doesn't mean it is interpritable.

Figure 37: You also sometimes don't care about interpritability. Examples above. Gaming the system means we can trick the system if we know how it works.

2.2 How can we interpret an existing ML model?

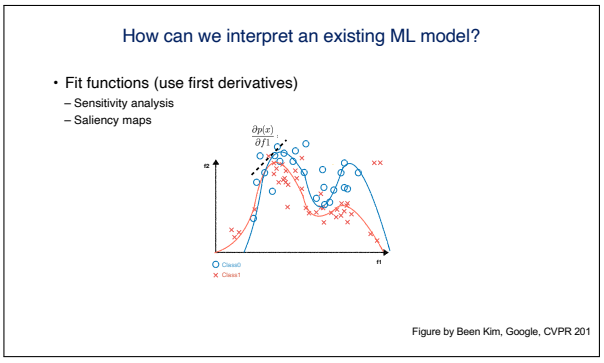


(a) One possible way is to say which feature is really important here? The abolation test removes a feature

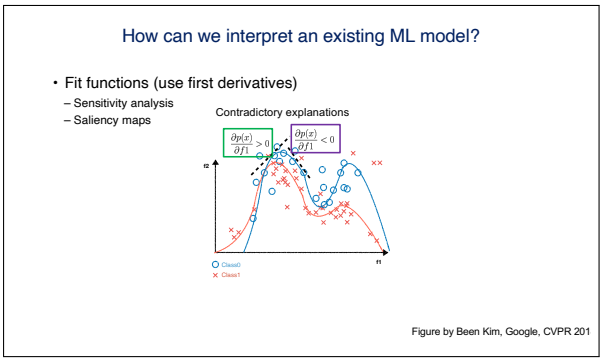


(b) this effectively projects it onto feature no.1 and thus is harder to classify becuase of the large overlap.

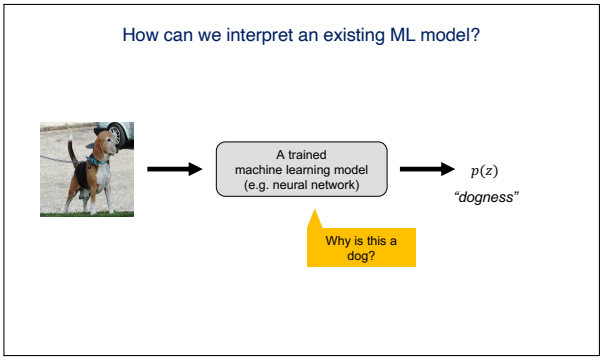
Figure 38: This doesn't scale well



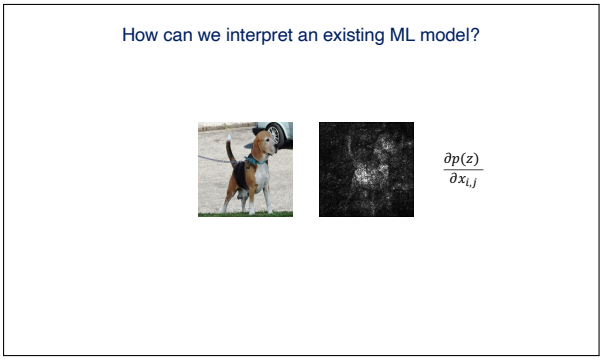
(a) Since we work with iamges, we can do sensitivity analysis or saliency maps. We take a particular sample and look at the gradient of the sample w.r.t feature1 and feature 2. This tells you how drastically if you cahnge the feature value how the classficiation will change.



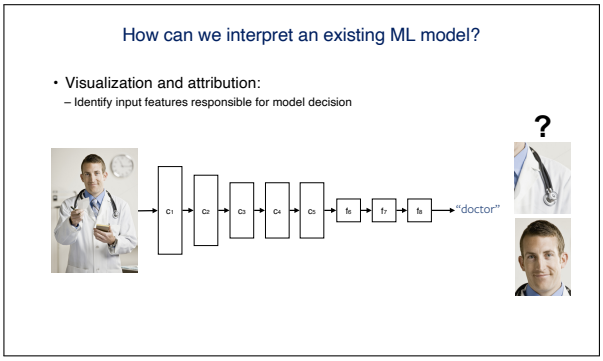
(b) If you do this on another value nearby you mgiht get a completely different answer which may give very contracting interpretations which are derrived based on similar samples



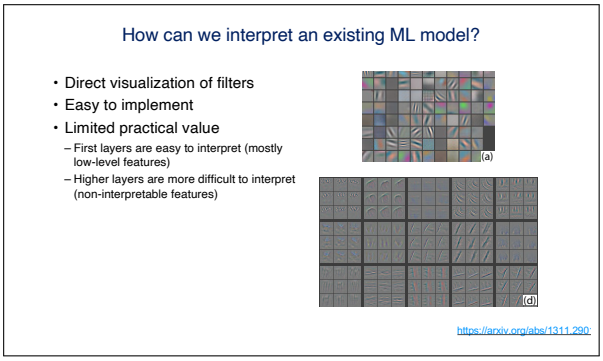
(c) A classifier predicts how much it looks like a dog. We want it to give an explination as to why its a dog.



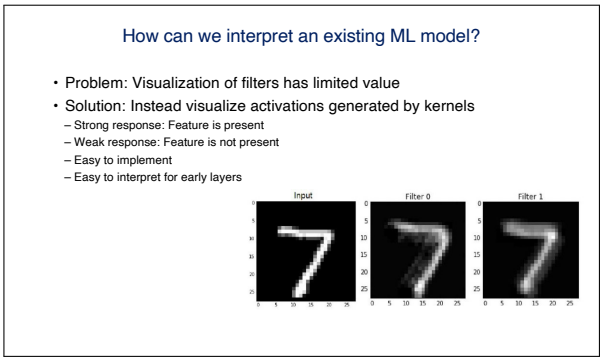
(d) In this example, sensitivity analysis gets an output like this. We compute the gradient on the output function w.r.t the input pixels (because these are teh features).



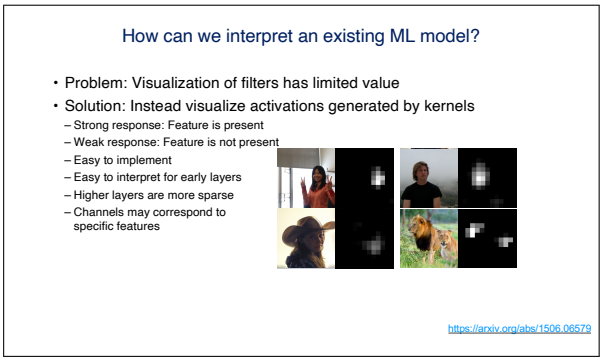
(e) With the stereotypical doctor image, i want it to reconzie reasons. Here, the top image is better. For example, the model may take a shortcut and predict that a man is more likley to be a doctor.



(f) How do we get to this? We can differentiate w.r.t the input pixels. Instead, we can visaulise the convolutional features. However, this is not interpretable



(g) A more practical approach is to focus on occlusion maps. We can give it a sample input and see what it outputs.



2.2.1 Occlusions

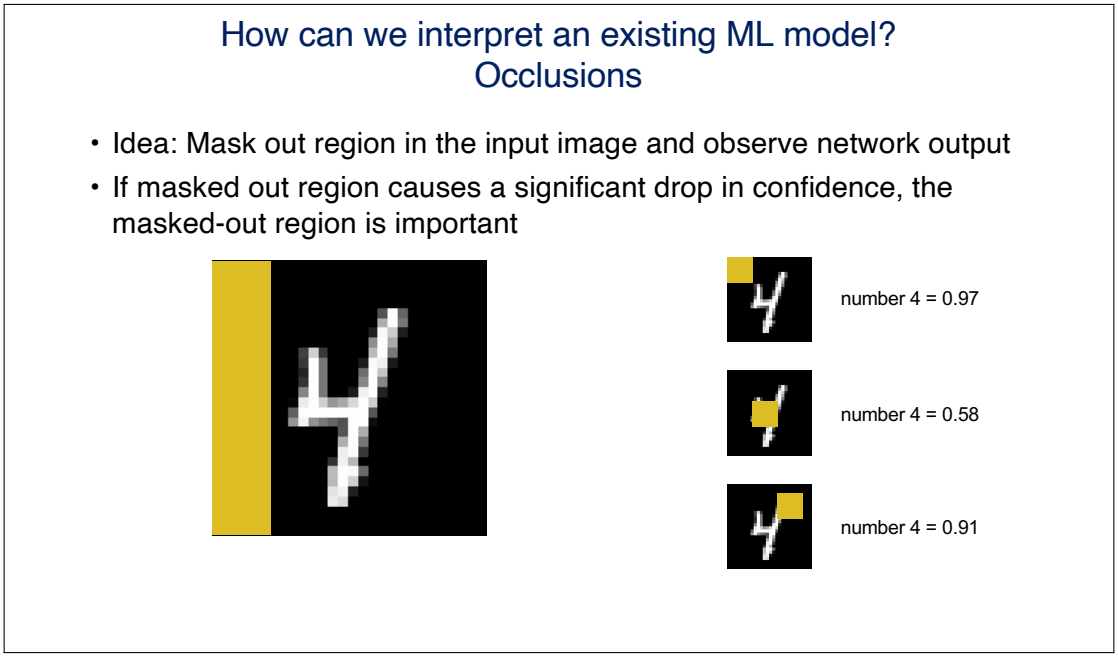
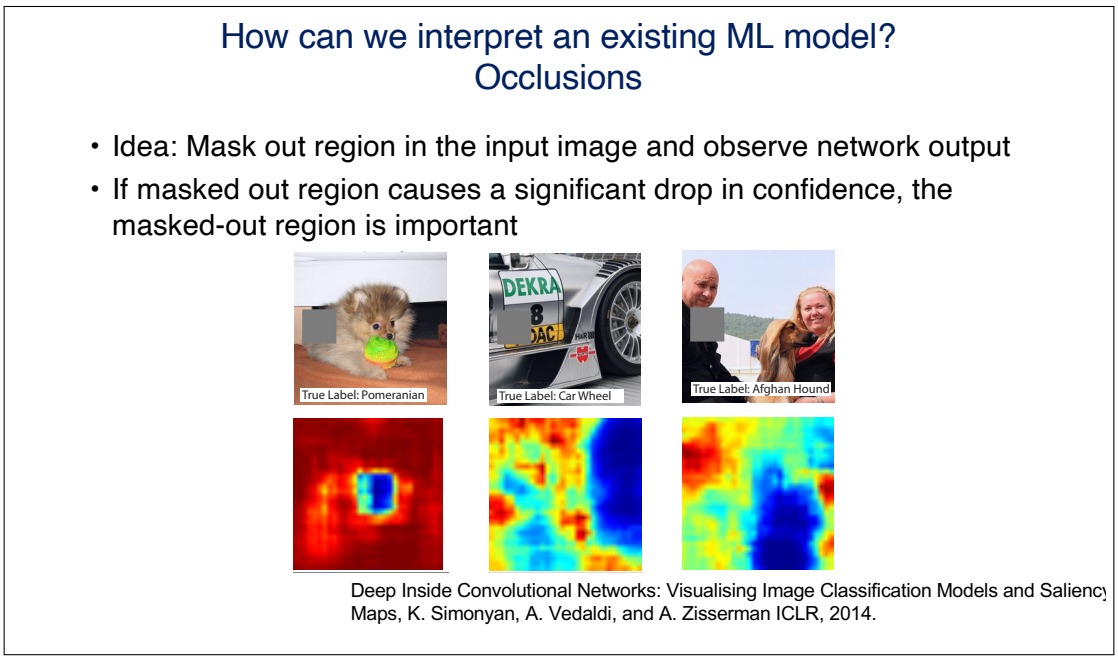


Figure 39: Here, slide an occlusion filter among the MNIST data. Here the model reacts more sensitively to some locations than others. To visalise this, we use a Salicency Map



2.2.2 Saliency maps

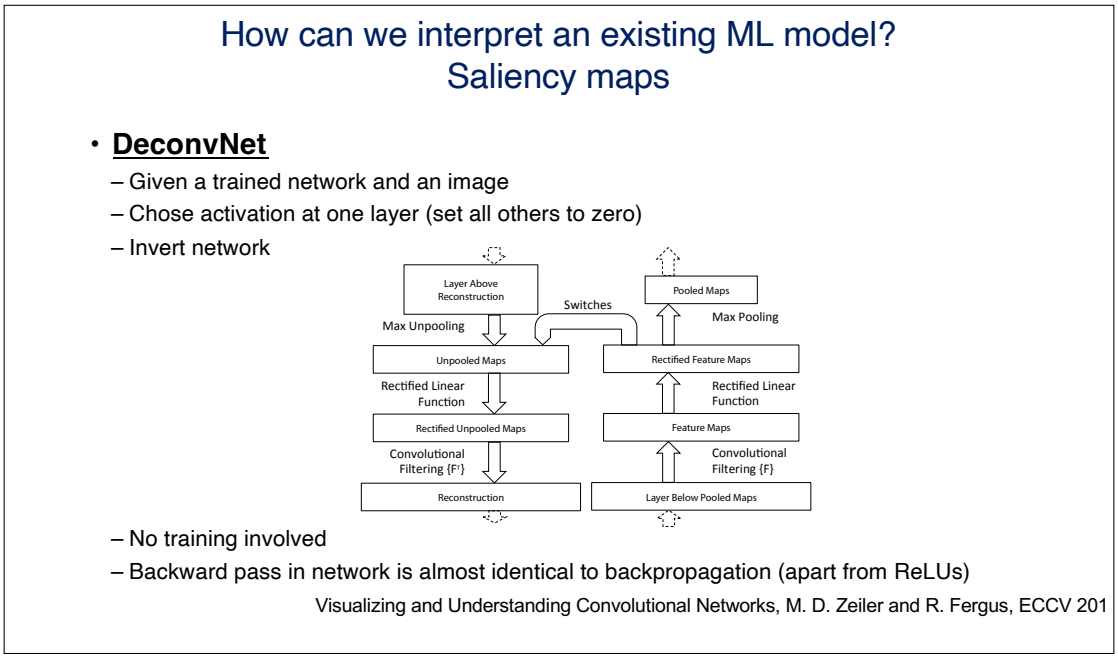


Figure 40: What this approach map did was try to invert the process of the forward pass. What you do now, is you take an image as input, you put it through the network up until a layer, then try to reverse it.

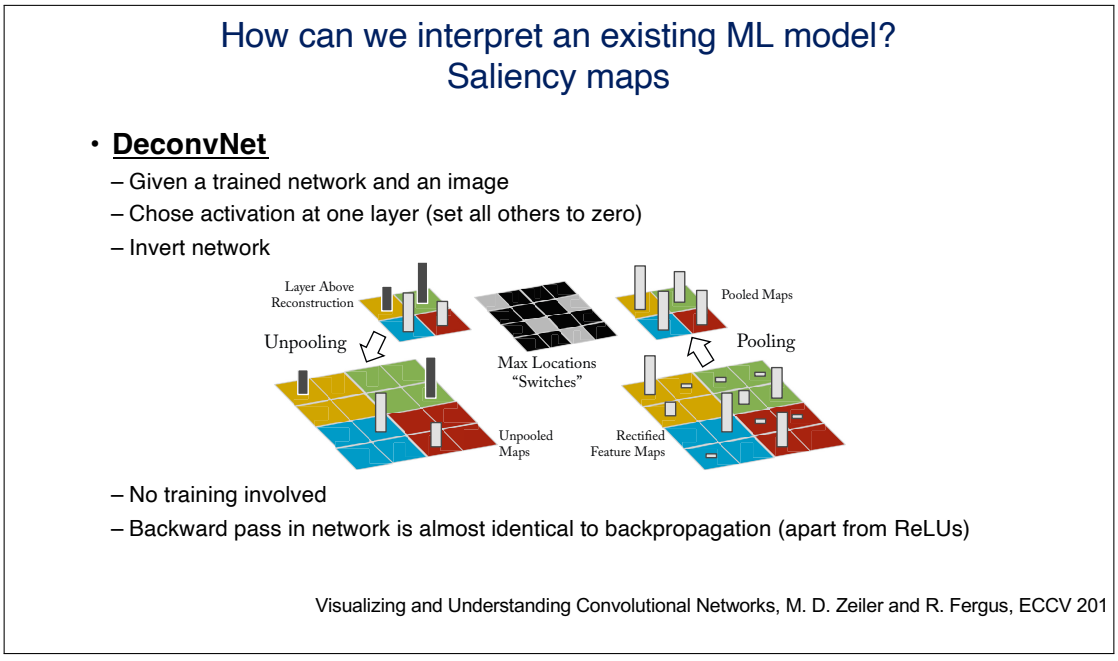


Figure 41: for each operation you need an inverse information (difficult since CNNs contract information). You do something similar as backpropogations, but in the forward pass you record infomration for the maximum locations. Then when you do back propogation you slap the values into where the max of the forward pass was.

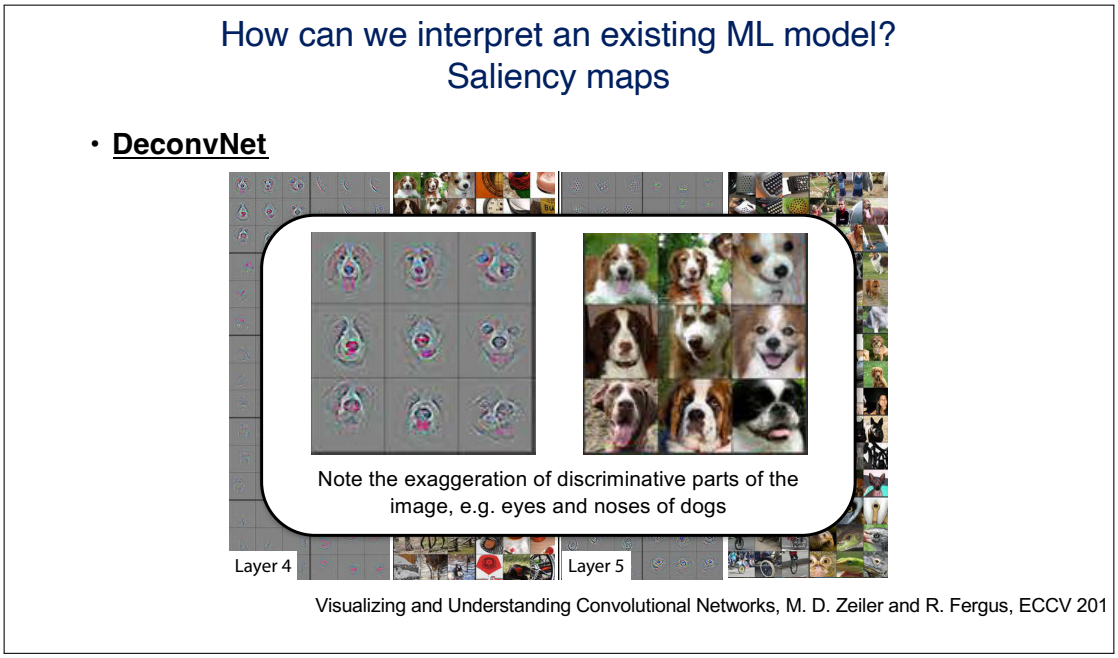


Figure 42: Here, there are many activations where the dogs face was activated. This allows you to investigate what each layer does.

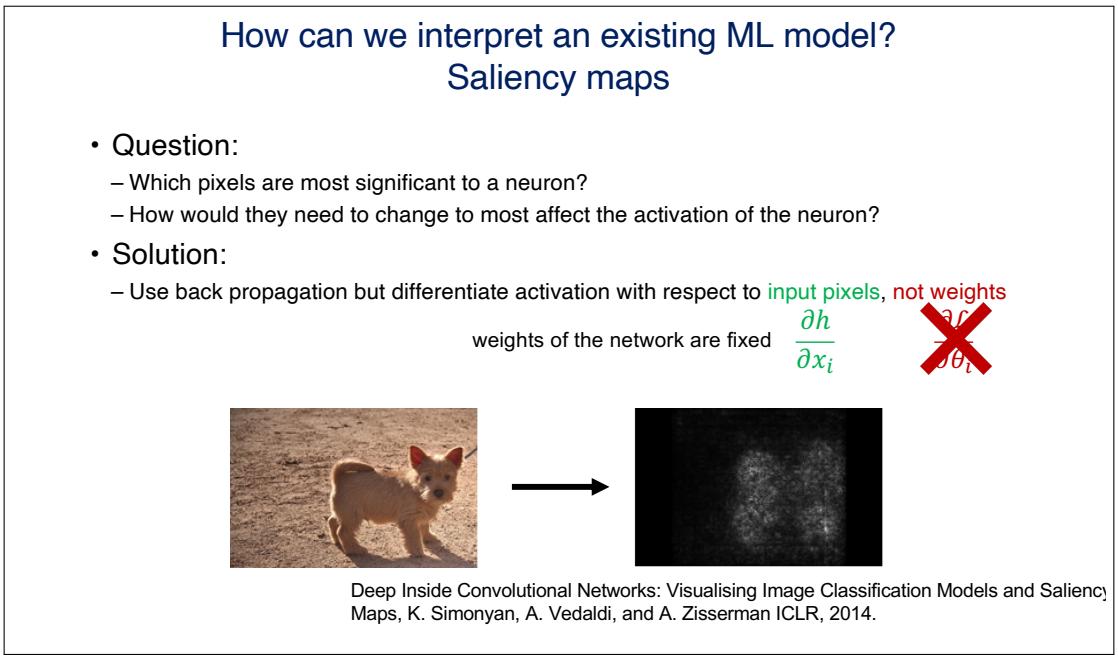


Figure 43: backpropogate with respect to input pixels (not the weights). But this isn't much more useful than just localising where the point of interest is.

2.2.3 Salicency maps - Gradient (backpropagation)

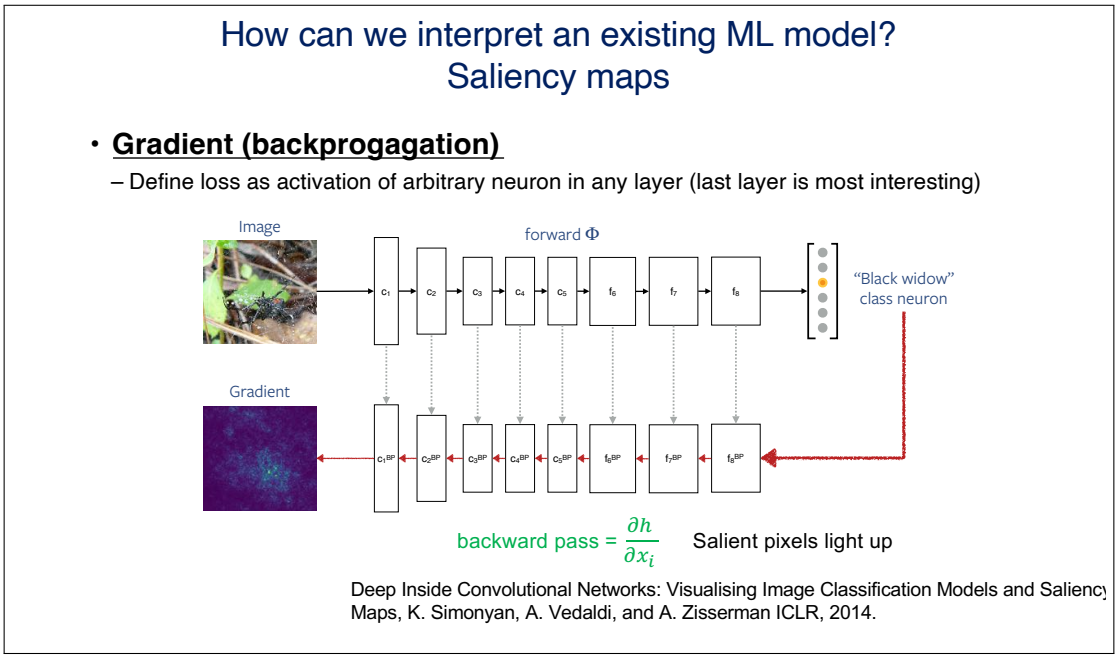


Figure 44: Simplest way is to do backpob: do the max acitvation and pretend that only this is avtivated and do your backpropoagation

2.2.4 Salicency maps - Guided backpropagation

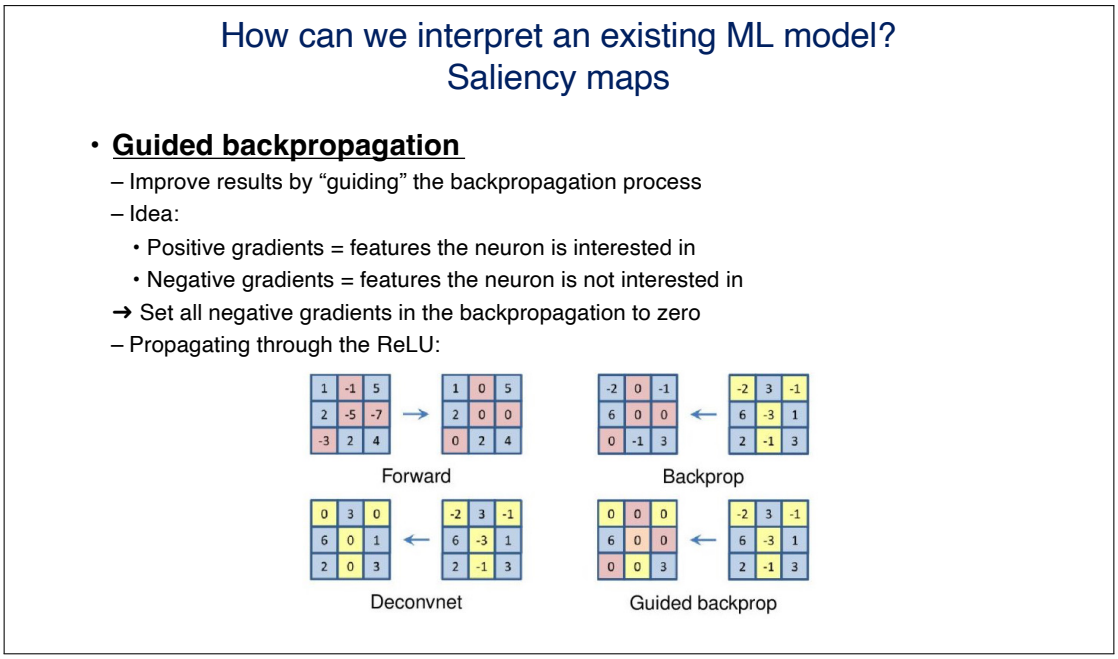


Figure 45: This make a simple modification; we are not interested in nerons that give negative gradients because negative gradients don't contribute anything to the output. The positive outputs represent features the neuron is interested in. This way, you set all the negative weights to zero.

Below, you see the three different schemes, standard backprop, deconvnet and guided backprop. What they differ in is how they deal with negative values.

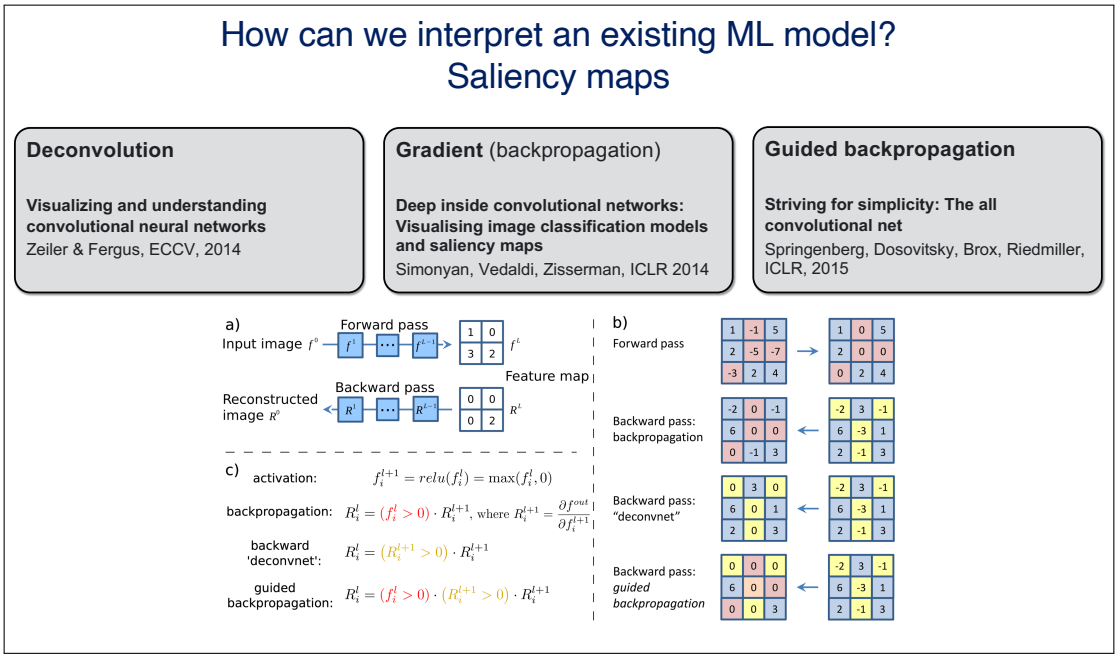


Figure 46: Rules for computing by hand

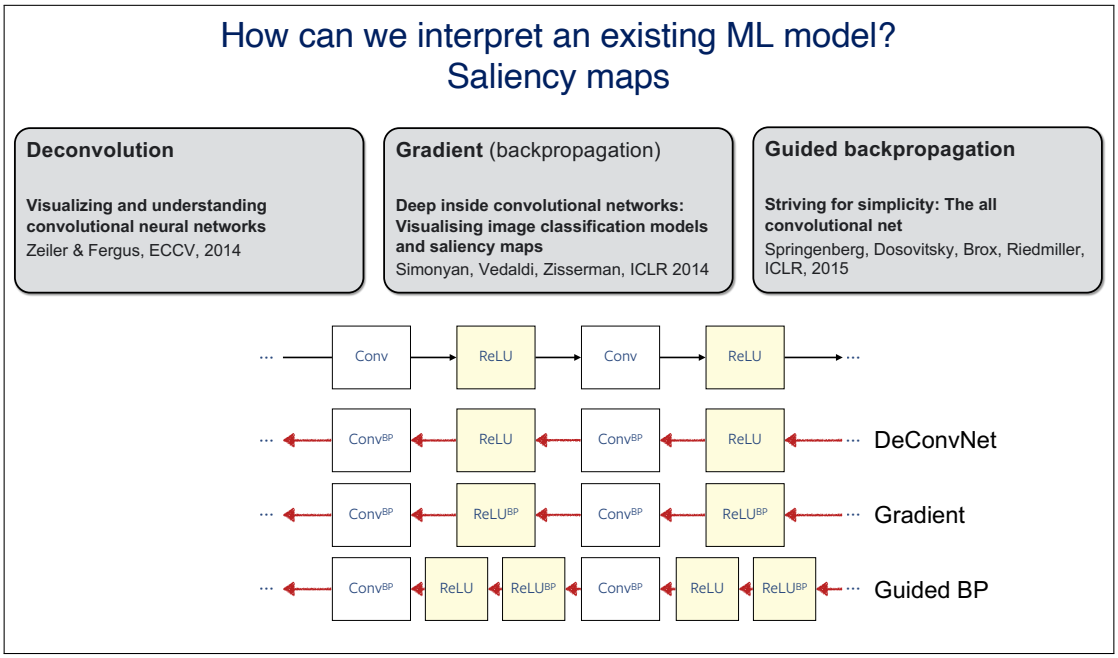


Figure 47

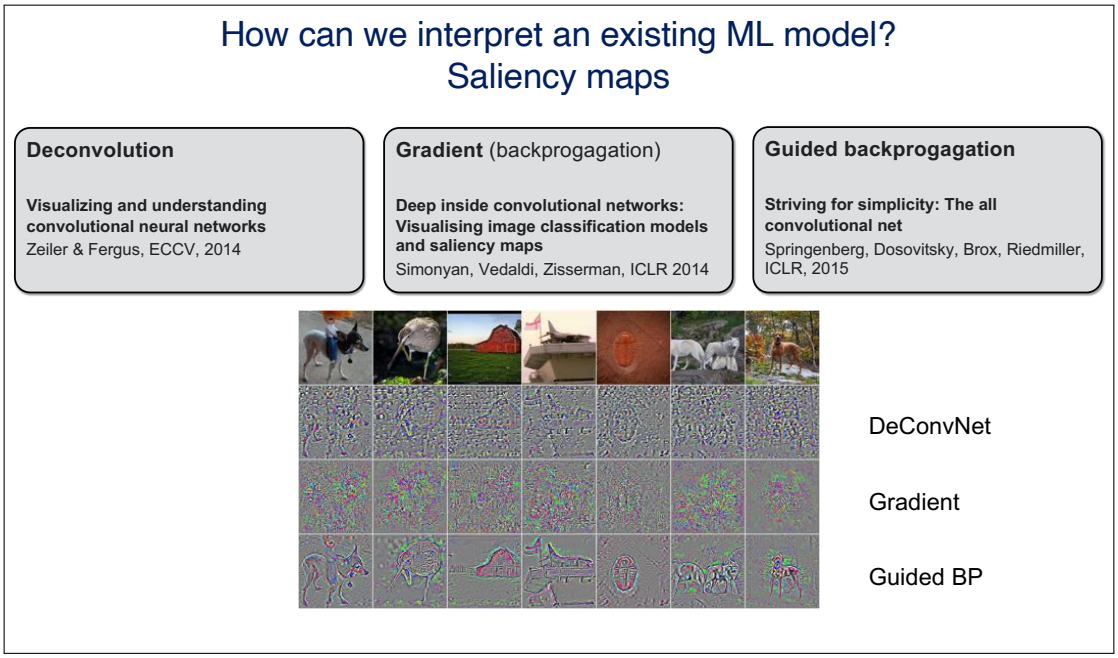


Figure 48: Guided backprob gives most meaningful results usually

2.3 Cam and Grad-Cam

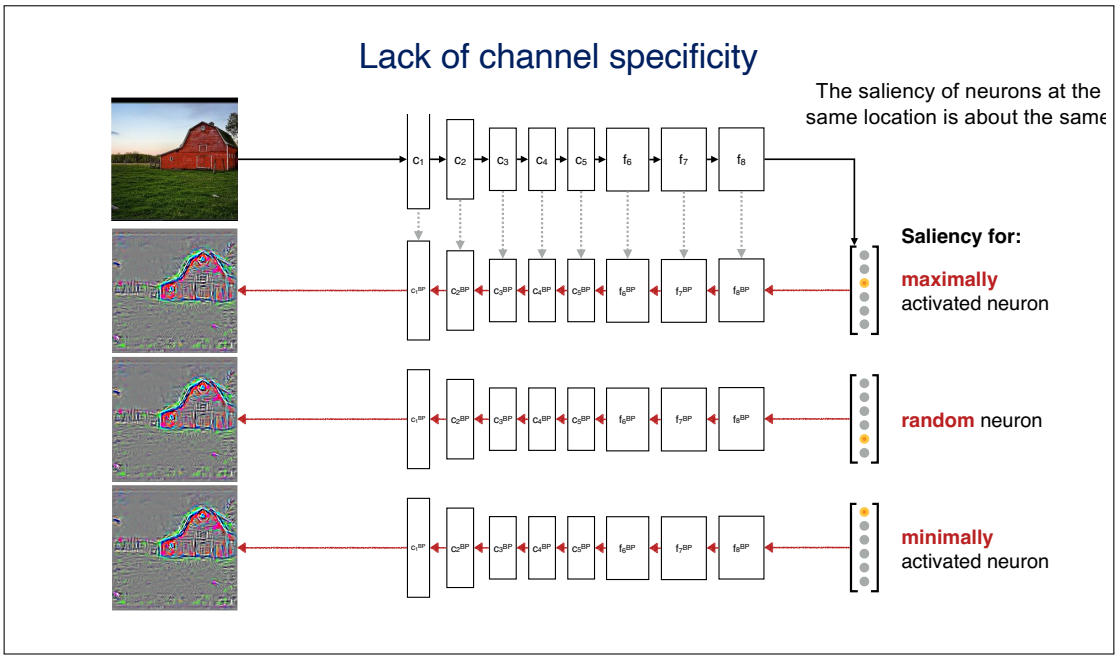


Figure 49: The saliency map should also change depending on the output of the network
if you have a saliency map fo rmaximally activated neruon and minimally activated neruon and they give the same map that doesn't make sense.

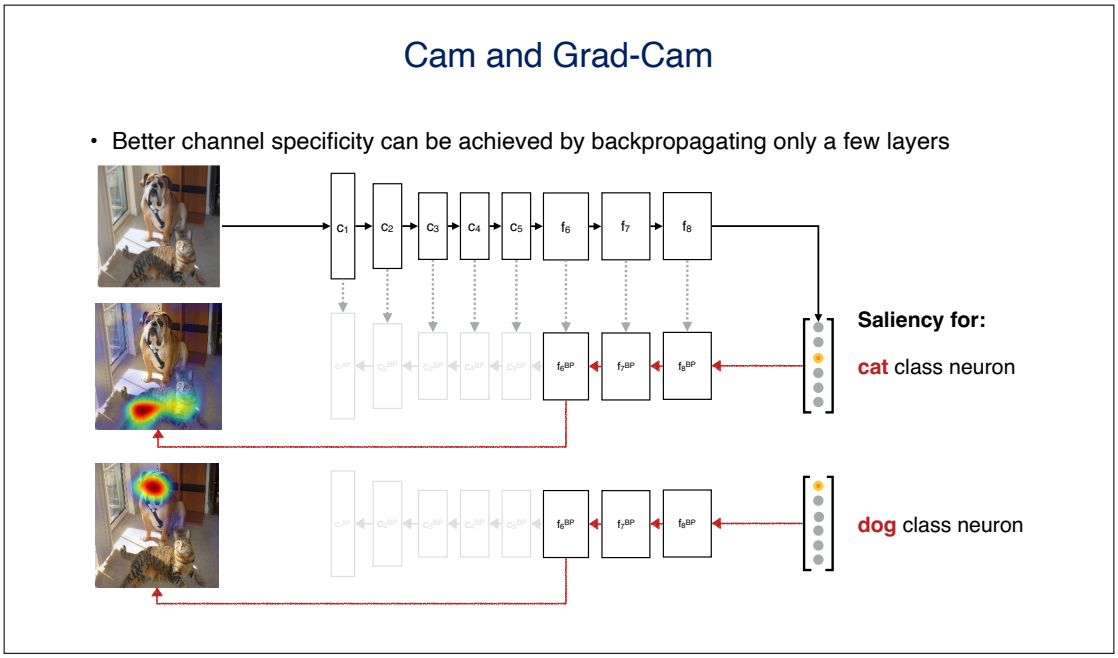


Figure 50: The Grad-Cam method allows you to query in much more detail what the differnet input channels are doing in terms of activation

2.4 DeepDream / Inceptionism

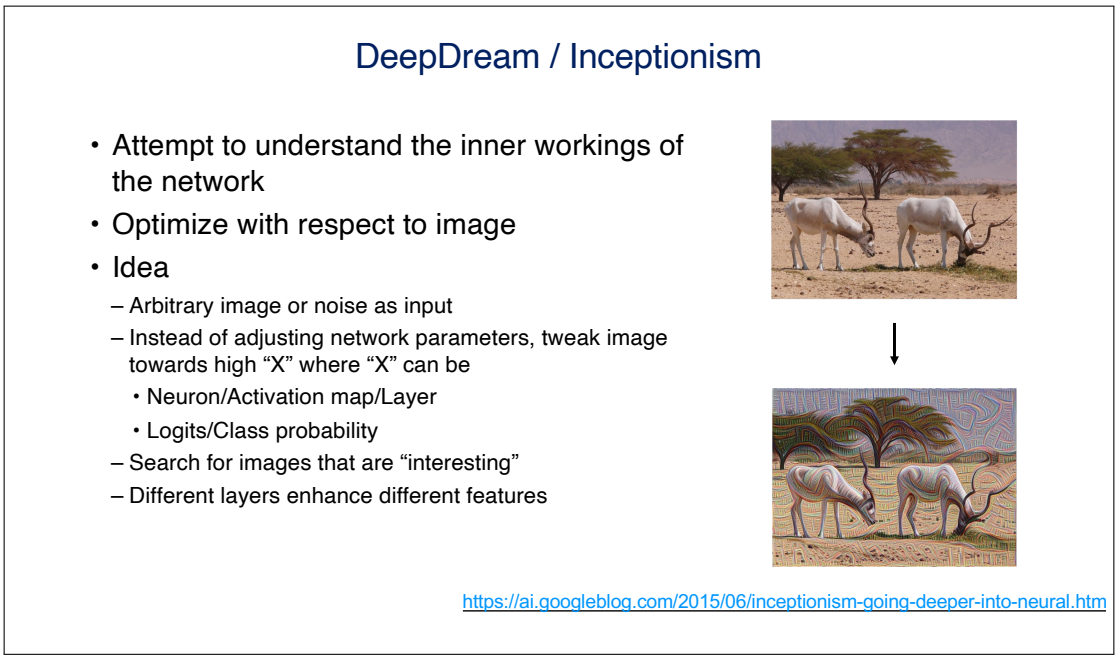


Figure 51: They use ideas of inverse problems to see what the networks are learning

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.htm

Figure 52: we want to find an image that has the same activation that oyu have already observed for an existing input image and you try to reverse engineer this.

2.5 Inversion

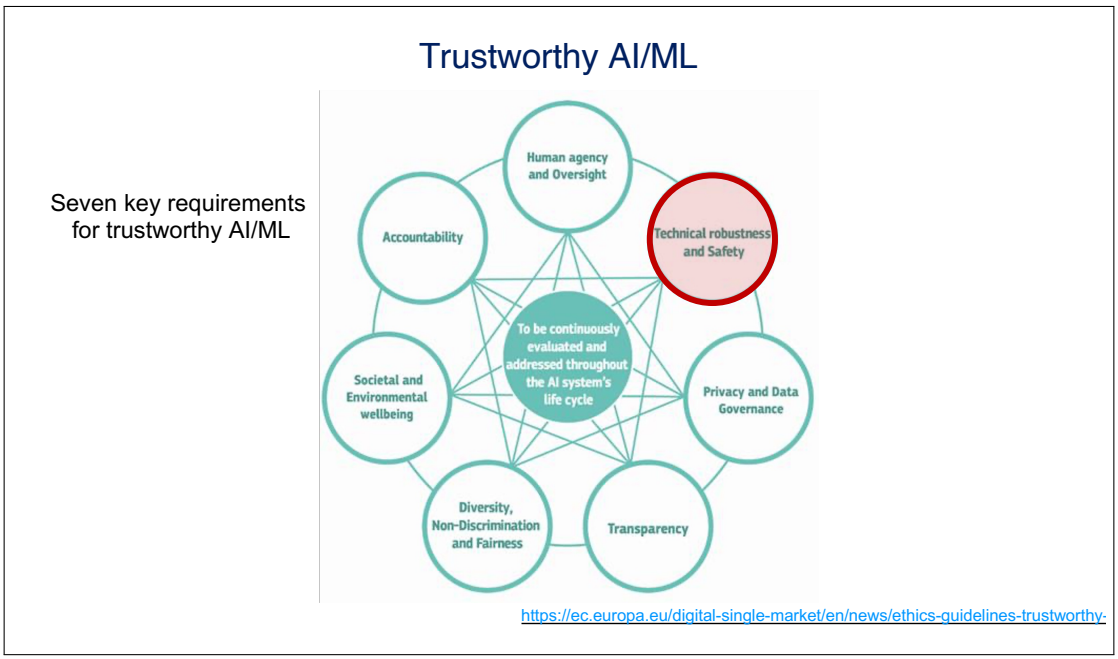
Inversion

- 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

3 Robustness: Adversarial Methods



3.1 Adversarial Attacks

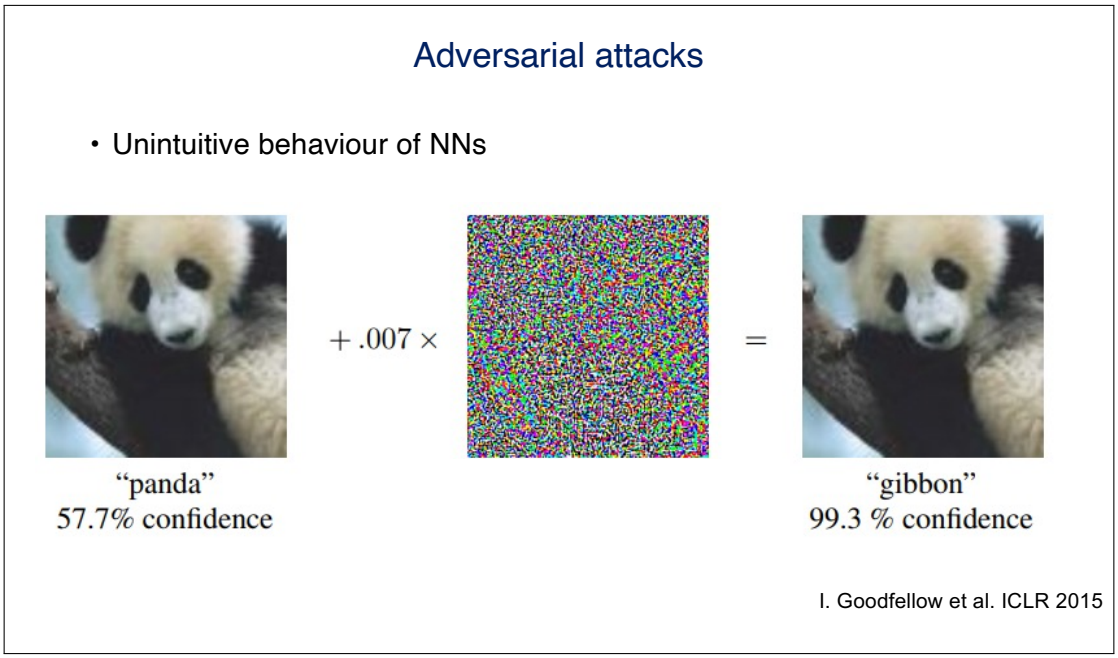


Figure 53: Two images have been produced; the difference between the two image is a small scaling factor of noise. The small difference shouldn't be enough to change the classification.

3.1.1 Pertubation

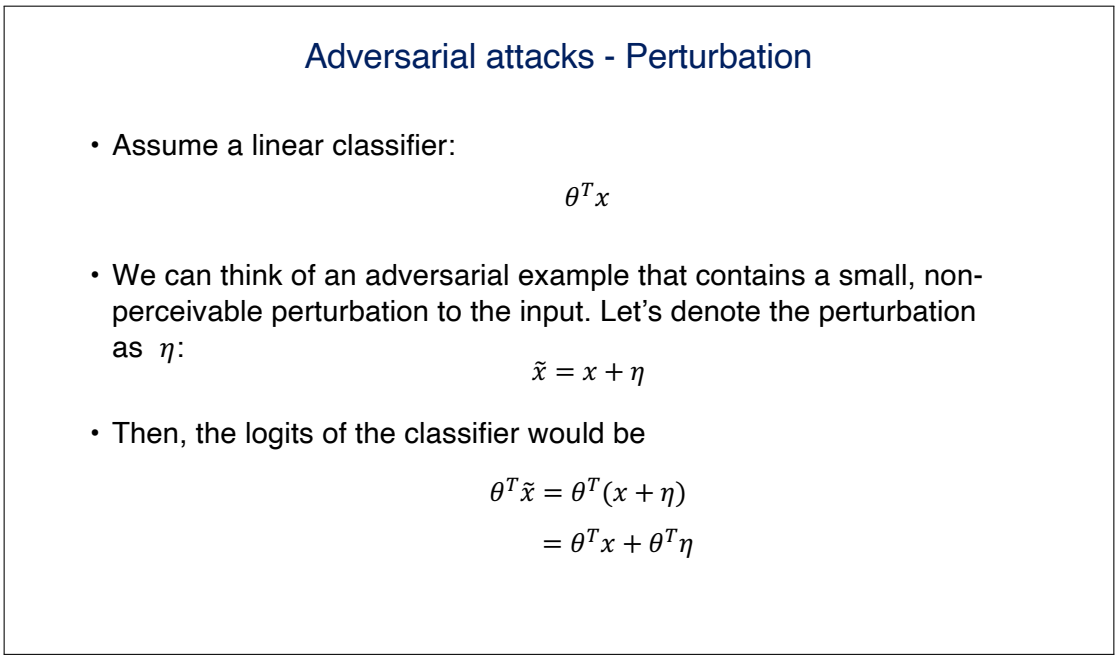


Figure 54: We add to x another vector and see what happens; we can't guarantee that despite the noise being small, the output will be the same.

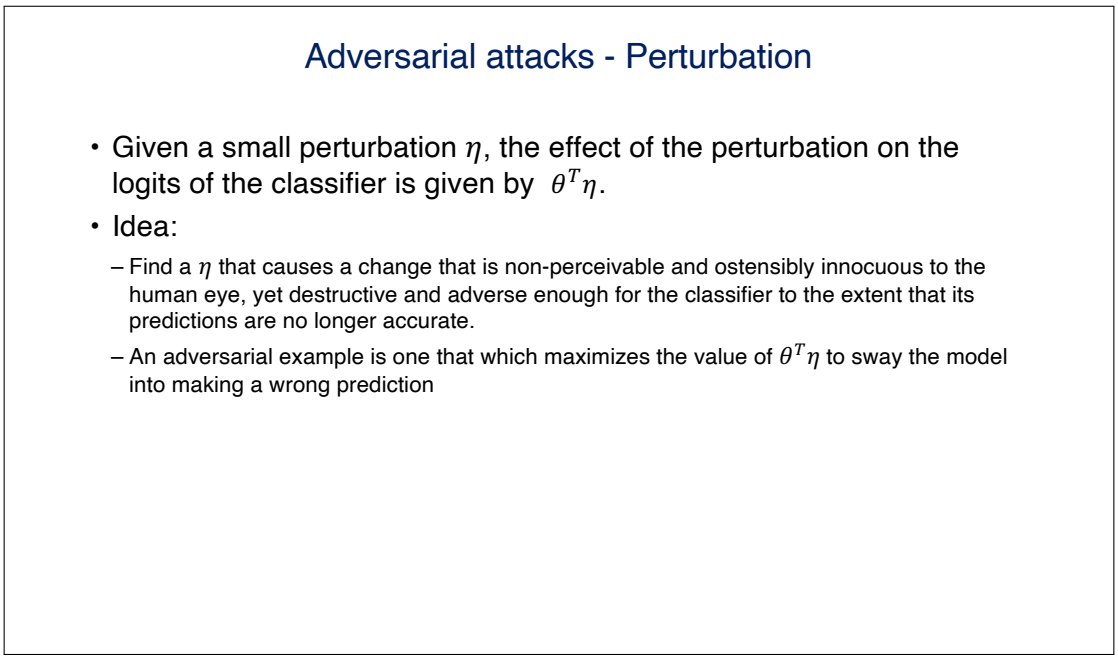


Figure 55: If you want to attack a model, you want to find a small pertubation that changes the classification drastically

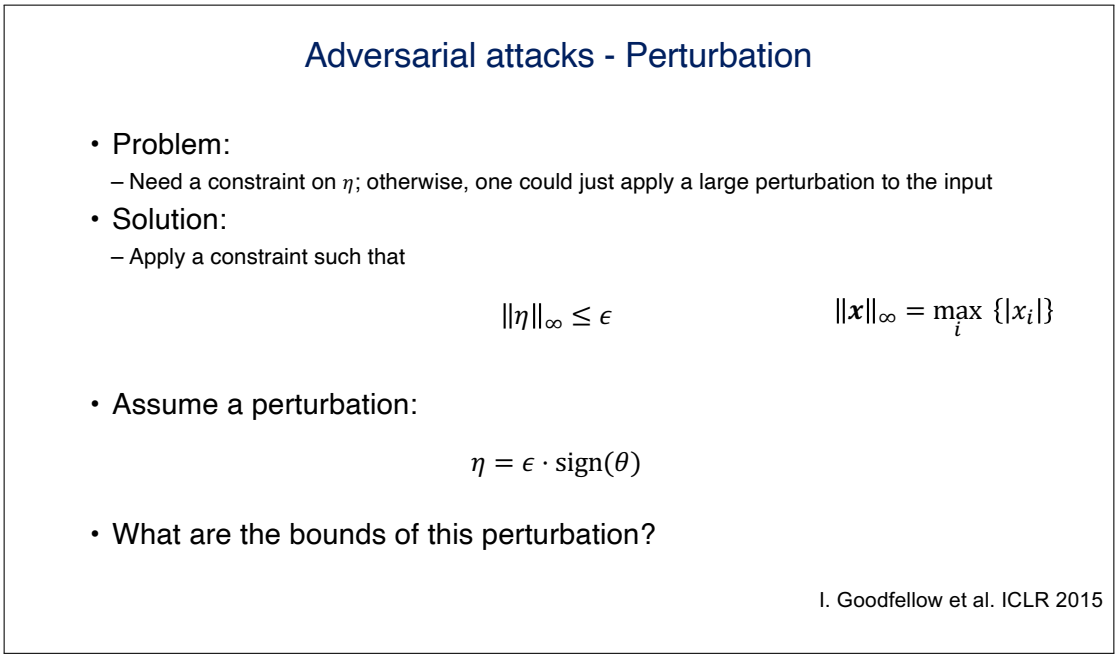


Figure 56: This is trivially easy to do with a large pertubation. However, we constrain it such that the largest element in the vector is less than epsilon (L_inf norm). Then we assume the samllest pertubation where the sign funciton produces eitehr 1 or -1.

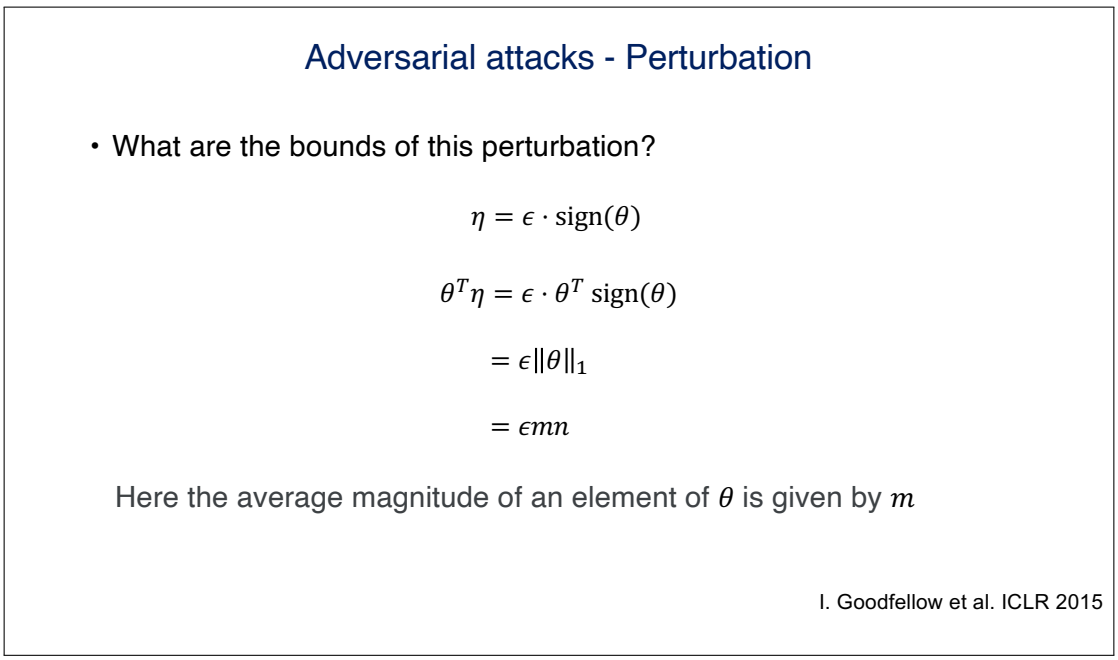


Figure 57: after some math we can find the magnitude of the vector.

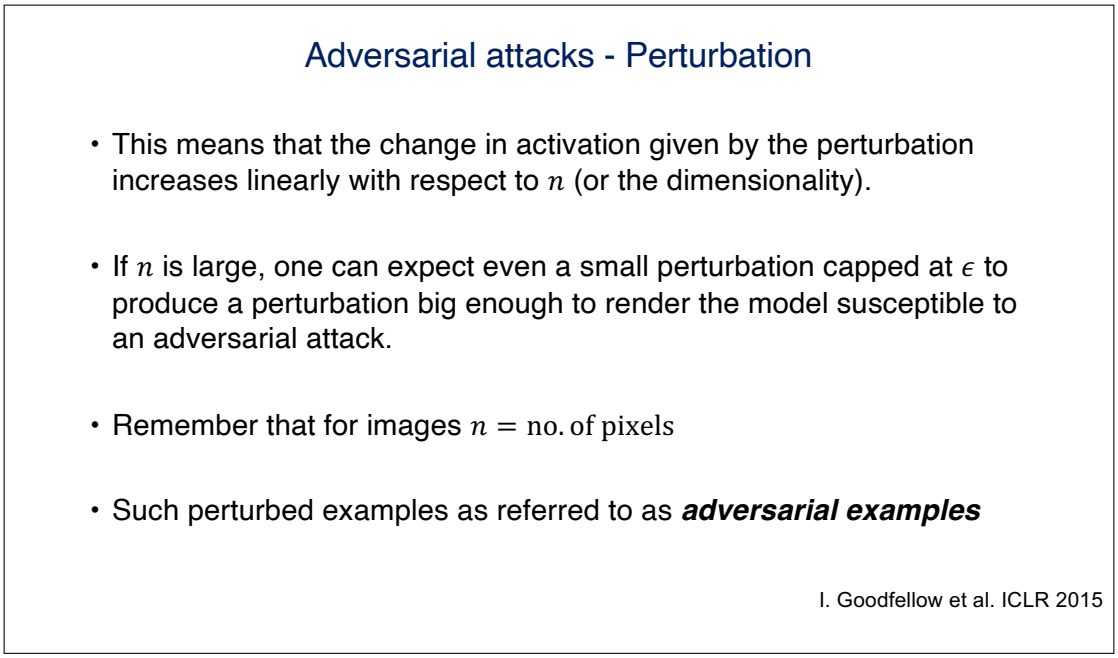


Figure 58: What this means is that as the dimensionality increases n gets very large. So this magnitude increases drastically with the number of pixels which means it becomes easier to find noise that will attack the model.

3.1.2 Fast Gradient Sign Method

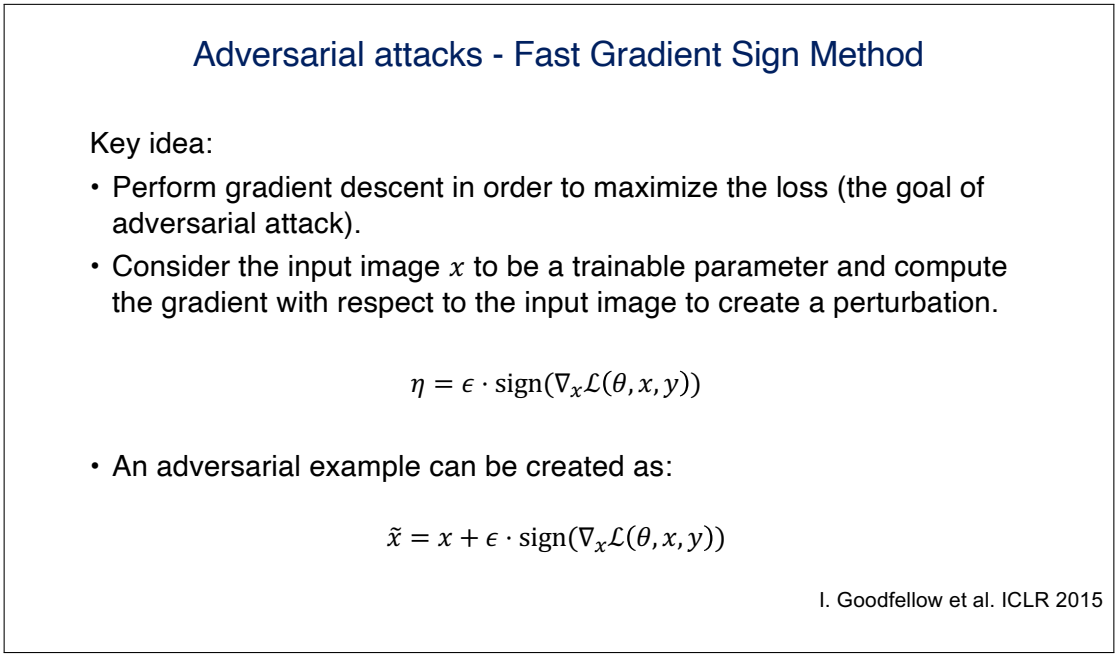


Figure 59: Now, you find the optimal vector that attacks the model, meaning that you add the gradient of loss function and then create this vector of zeros -1s and 1s depending on which direction the gradient points, you can create an adversarial example very easily.

3.2 Adversarial attacks

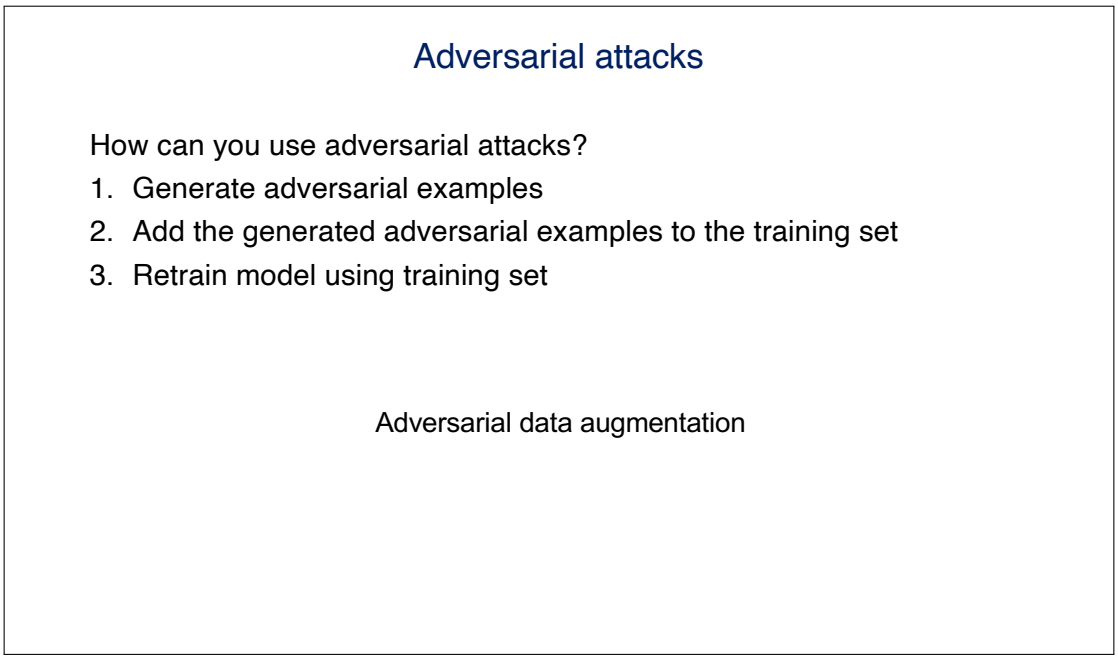


Figure 60: You can't defend that easily.

A References

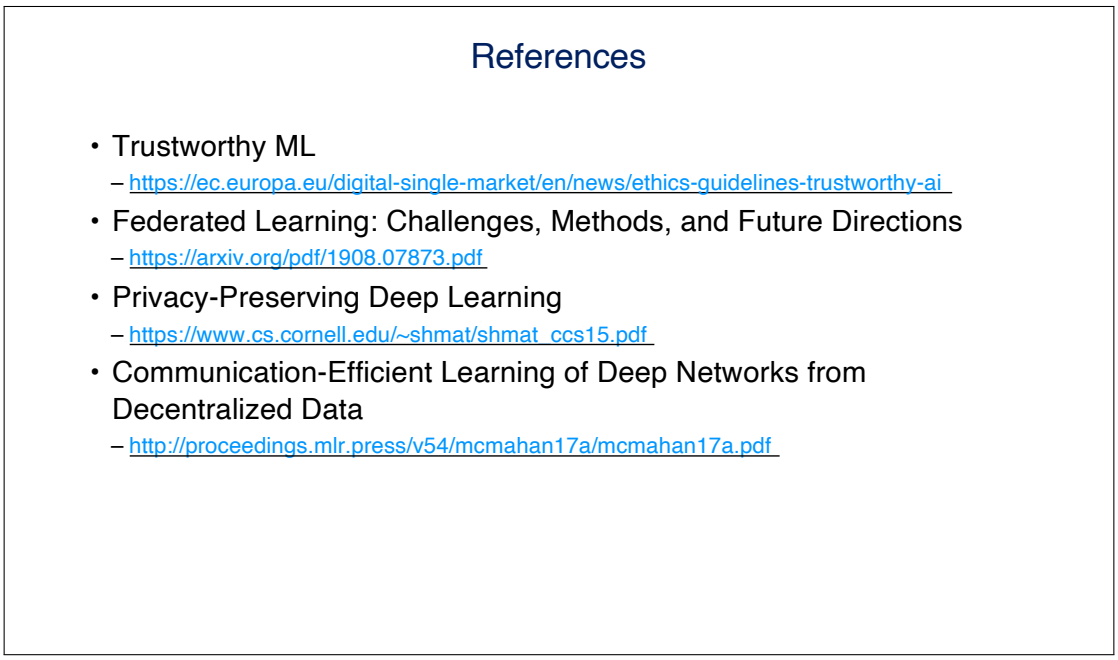


Figure 61: caption

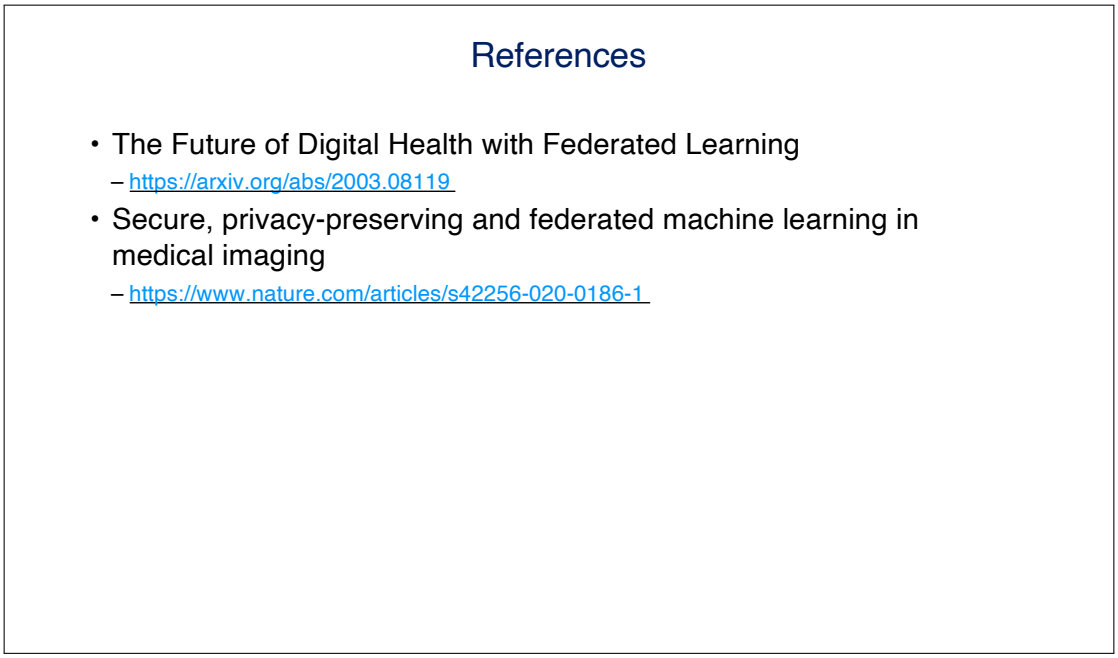


Figure 62: caption

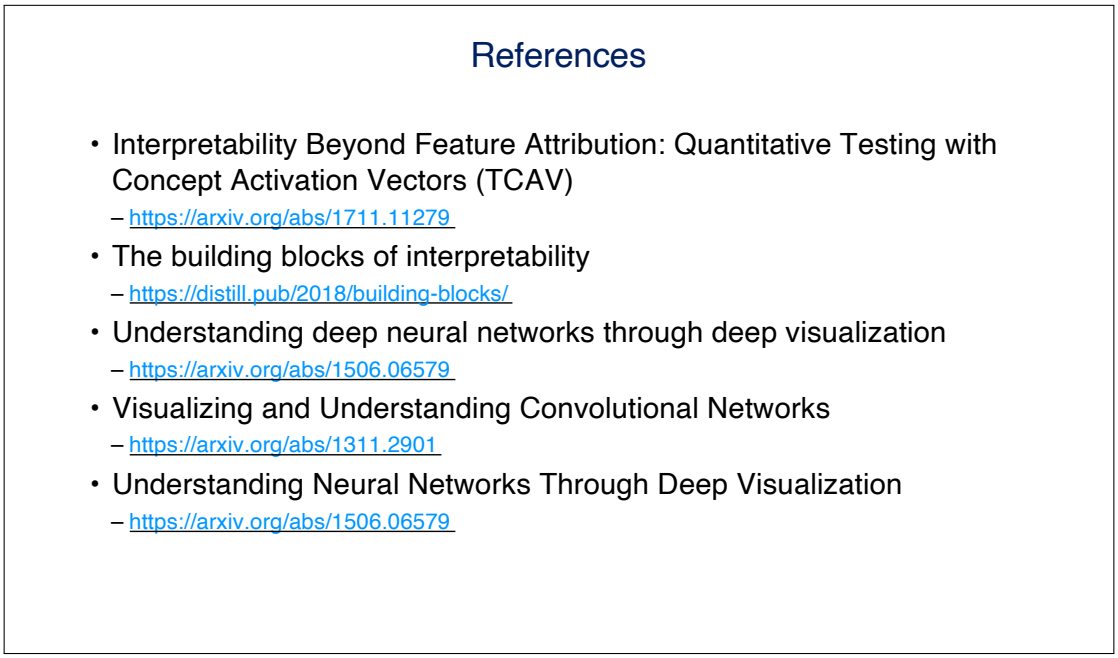


Figure 63: caption

References

[1] Dmitrii Usynin et al. “Adversarial interference and its mitigations in privacy-preserving collaborative machine learning”. In: *Nature Machine Intelligence* 3.9 (Sept. 2021), pp. 749–758. ISSN: 2522-5839. DOI: 10.1038/s42256-021-00390-3. URL: https://doi.org/10.1038/s42256-021-00390-3.