Ahmed Ibrahim

ECE 9039/9309 MACHINE LEARNING

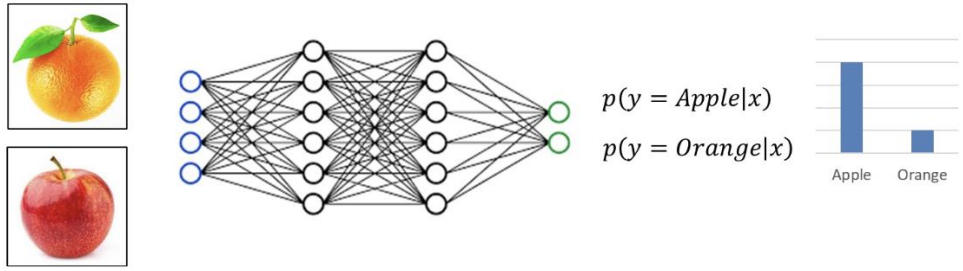
Outlines

- Reliability and the ethical considerations related to machine learning predictions
- Federated Learning
- Intro. To Generative AI

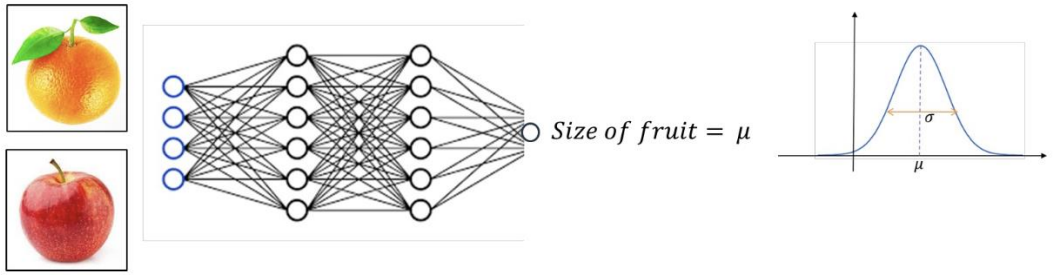
Reliability in ML Predictions

Confidence Score

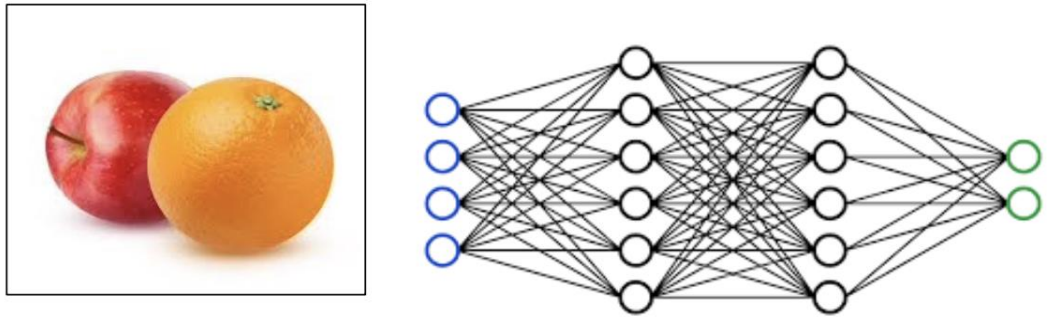
Classification



Regression



- What if?



$p(y = Apple|x)$ = 0.45

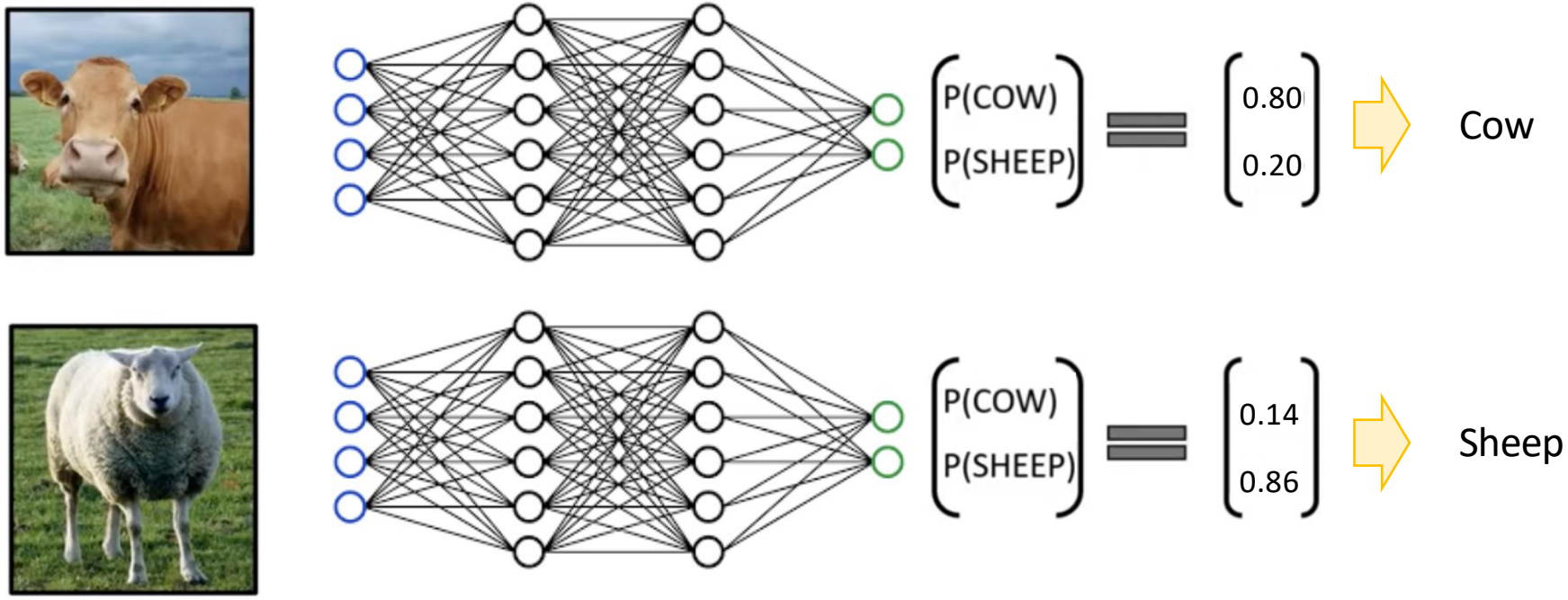
$p(y = Orange|x)$ = 0.55

- The output will be <u>unreliable</u> if the testing data differs from the training data.
- The **confidence score** represents the probability (or a similar metric) that the given decision is correct, based on the model's learned patterns and the data it was trained on.

Model Confidence

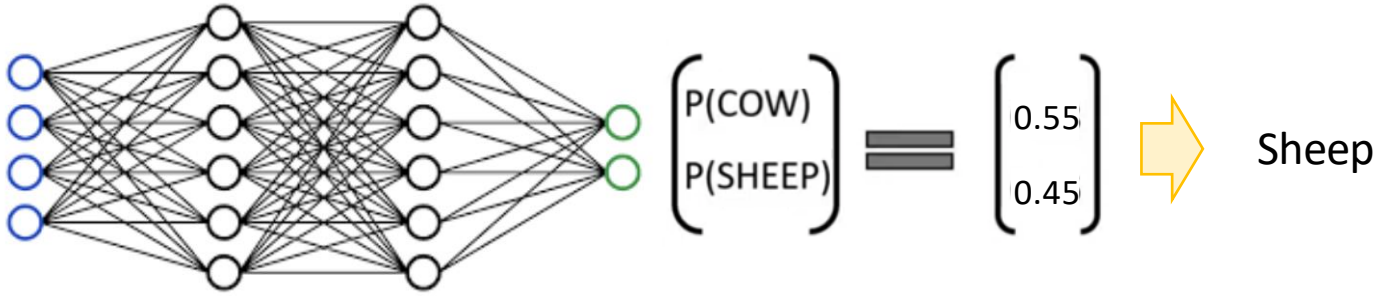
- Model confidence refers to the **model's own assessment** of how **certain** it is about its predictions.
  - For instance, in a classification task, a model might predict that a given image contains an apple with 90% probability. This probability reflects the **model's confidence** in its prediction based on the data it has been trained on and its learned parameters.
  - For regression models, confidence can be expressed in intervals (**confidence intervals**) that likely contain the estimated parameter's true value.
- **Pros**: Offers a straightforward, quantifiable measure of how much the model "believes" in its prediction.
- **Cons**: Confidence scores can be misleading if poorly calibrated, as a model might exhibit high confidence in incorrect predictions.

Another Classification Example



Uncertainty

Young Cow



- **Uncertainty** indicates the model's prediction limitations due to **inherent data noise** or **gaps in the model's knowledge**.

Types of Uncertainty

Uncertainty can be divided into two main types:

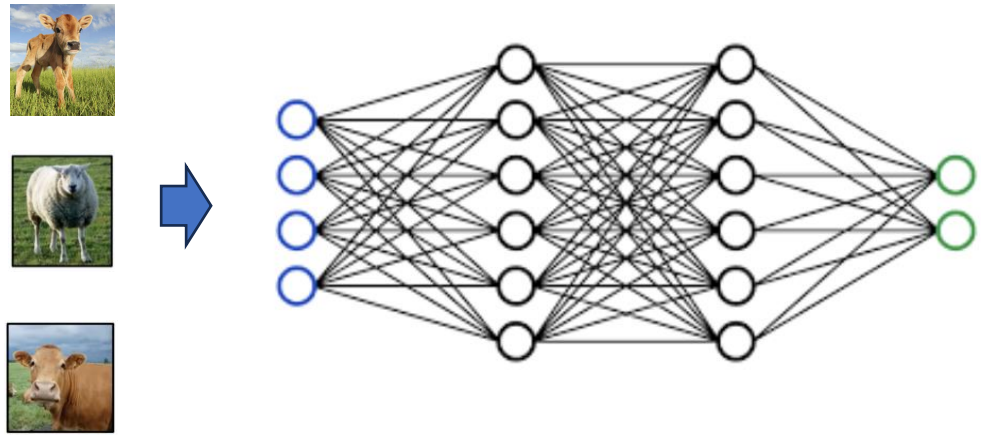
- Data **(Aleatoric or Natural)** Uncertainty comes from the noise inherent in the data and cannot be reduced even if more data is collected. For example, sensor noise or inherent ambiguities in the data contribute to aleatoric uncertainty.
- Model **(Epistemic or Lack of Knowledge)** Uncertainty arises from the model's lack of knowledge. It reflects the model's training limitations, such as having limited training data or encountering data significantly different from the training set.
- Unlike aleatoric uncertainty, epistemic uncertainty can be reduced by gathering more data or improving the model.

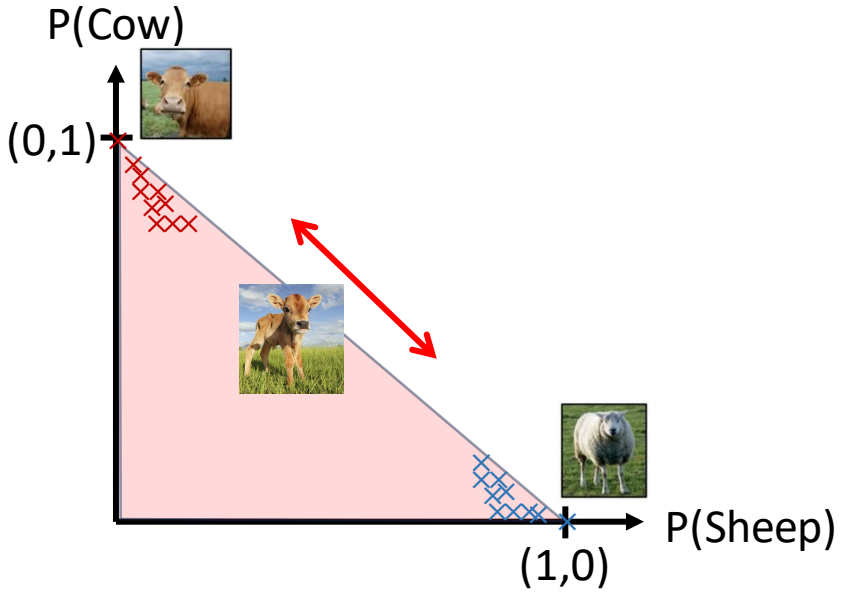
- Total Uncertainty = Data Uncertainty (Aleatoric) + Model Uncertainty (Epistemic)

Uncertainty (cont.)

- **Pros:** Provides insight into the **reliability** of predictions beyond confidence scores, indicating areas where the model's predictions should be taken cautiously.
- **Cons:** Quantifying uncertainty, especially model uncertainty, can be challenging. While advanced techniques exist, they're not always easy to implement.

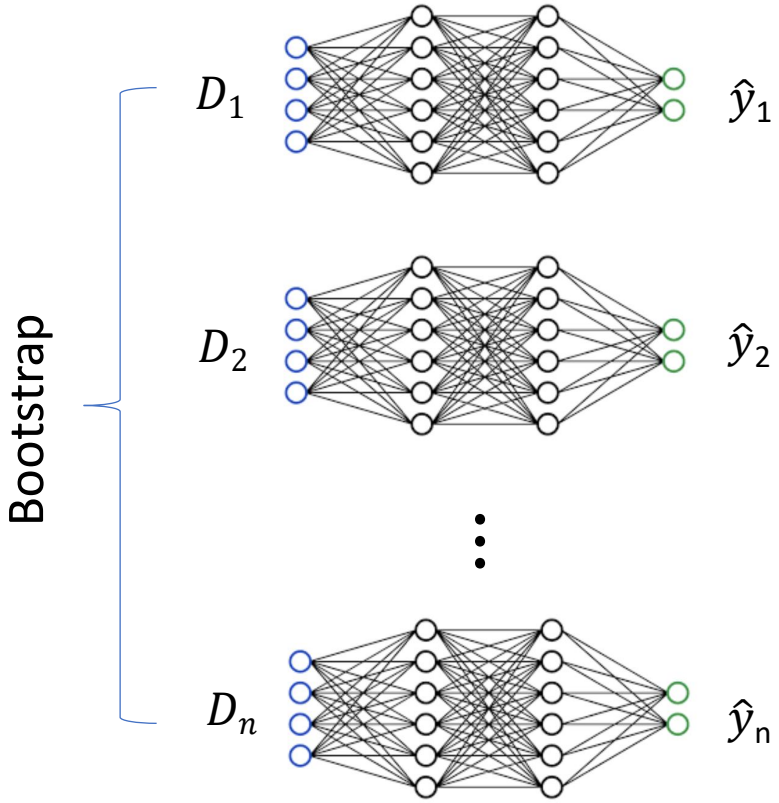
Uncertainty (cont.)

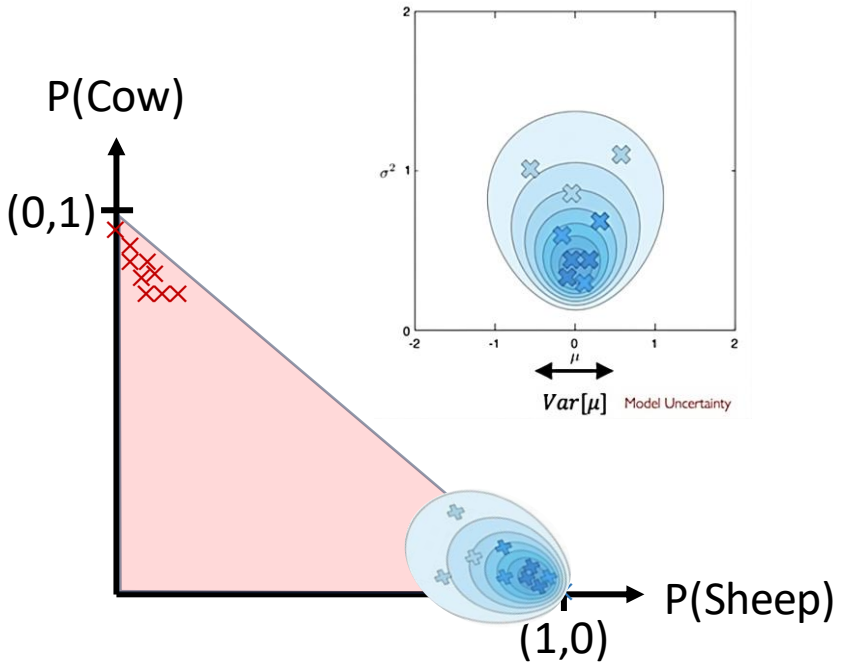


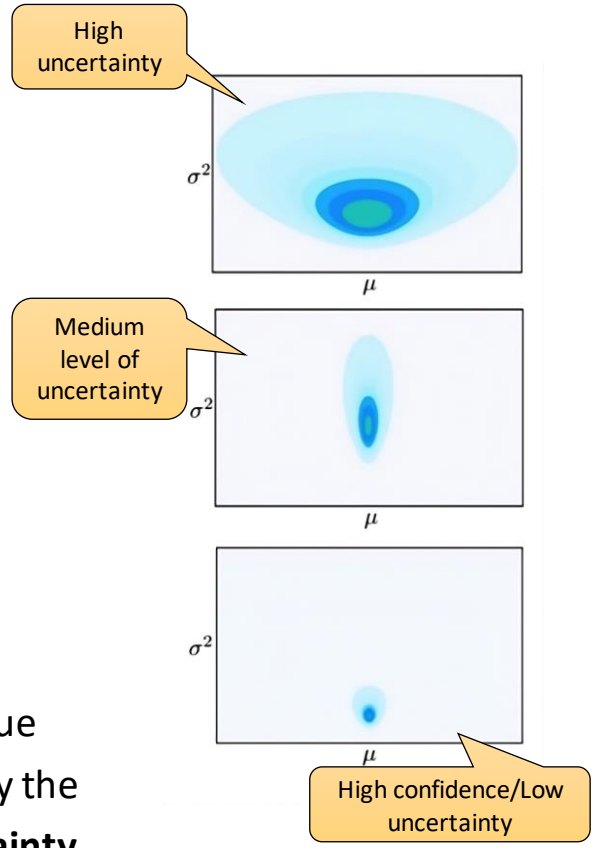


Uncertainty (cont.)

Ensemble:







- The center of the curve indicates the most likely value (mean), and the width of the curve, characterized by the standard deviation, reflects the high level of **uncertainty**.

Factors Affecting Uncertainty

- The quality of the training data.
- The **inherent variability** in the data. Some data sets are more predictable than others.

Data

- The complexity of the model. Too simple a model might not capture all the nuances of the data (underfitting), while too complex a model might start capturing noise as if it were a genuine pattern (overfitting).

Model

Biases in ML

Fairness Issues - Biases in ML

- What do you see?
  - Bananas
  - Stickers
  - Dole Bananas
  - Bananas at a store
  - Bananas on shelves
  - Bunches of bananas

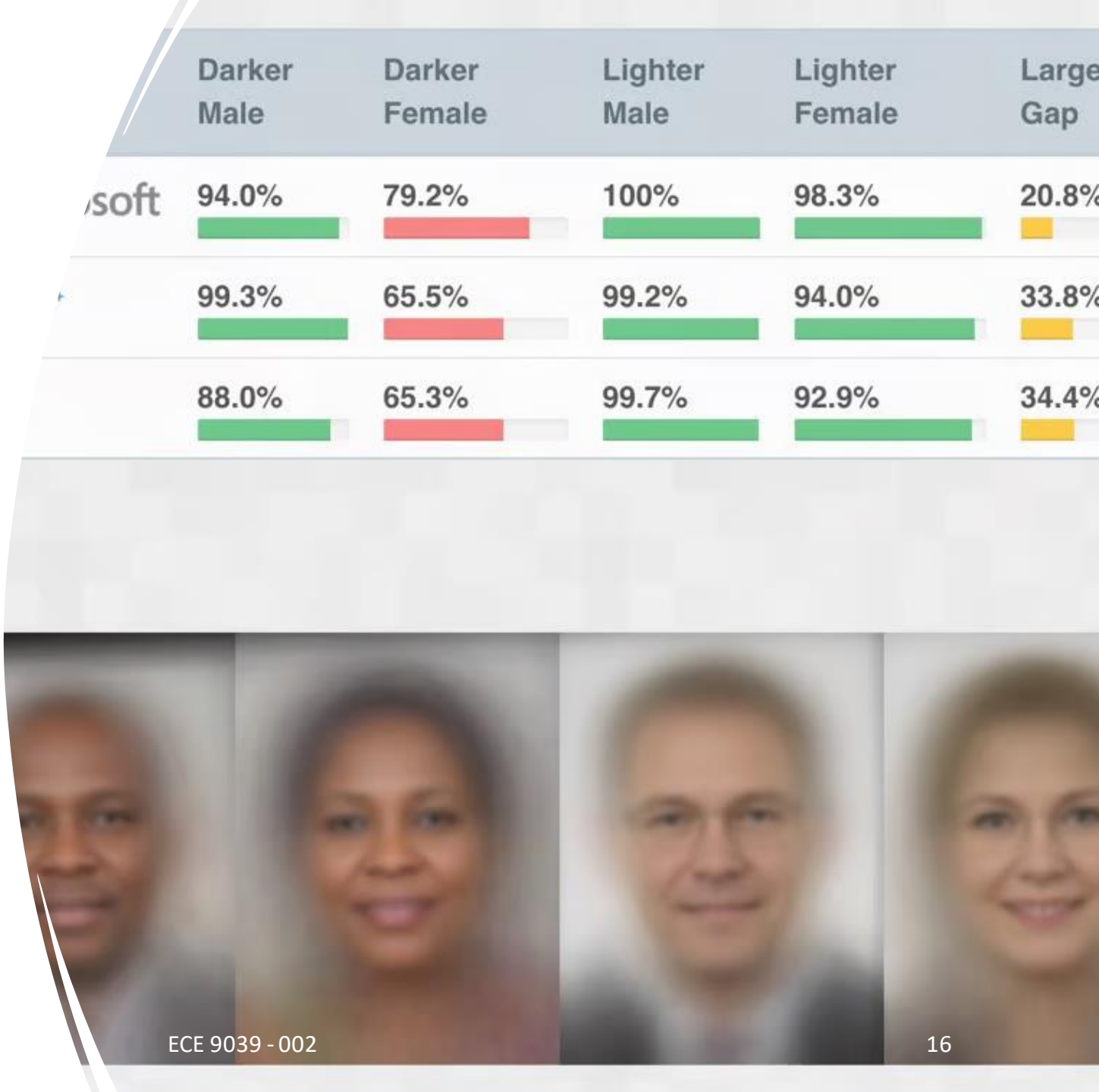
- ...We don't tend to say Yellow Bananas!!

Fairness Issues - Biases in ML

- What do you see?
  - Green Bananas
  - Unripe Bananas

Biases in ML(Race and Gender Diversity)

- “Facial recognition is accurate if you’re a white guy” The New York Times Feb 9, 2018
- In the Gender Shades (2018) project, Buolamwini found that face analytics systems **misclassify** darker-skinned women at a higher rate (20% - 35%) compared to other groups.



Income Predictions Example

- Predict income level based on the features provided

Prediction $\hat{y} = \begin{cases} 1 \rightarrow \text{loan} \\ 0 \rightarrow \text{no loan} \end{cases}$

$\text{race} = \begin{cases} 1 & \text{if White} \\ 0 & \text{if Black, Asian-Pac-Islander, Amer-Indian-Eskimo or Other} \end{cases}$

	age	education-num	marital-status	occupation	hours-per-week	native-country	race	sex	y
0	39	13	Never-married	Adm-clerical	40	United-States	White	Male	<=50K
1	50	13	Married-civ-spouse	Exec-managerial	13	United-States	White	Male	<=50K
2	38	9	Divorced	Handlers-cleaners	40	United-States	White	Male	<=50K
3	53	7	Married-civ-spouse	Handlers-cleaners	40	United-States	Black	Male	<=50K
4	28	13	Married-civ-spouse	Prof-specialty	40	Cuba	Black	Female	<=50K
5	37	14	Married-civ-spouse	Exec-managerial	40	United-States	White	Female	<=50K
6	49	5	Married-spouse-absent	Other-service	16	Jamaica	Black	Female	<=50K
7	52	9	Married-civ-spouse	Exec-managerial	45	United-States	White	Male	>50K
8	31	14	Never-married	Prof-specialty	50	United-States	White	Female	>50K
9	42	13	Married-civ-spouse	Exec-managerial	40	United-States	White	Male	>50K

- A survey that collects various pieces of demographic and employment-related information from individuals.

Measuring Fairness – Metrics (cont.)

- **Accuracy** – Measures whether accuracy is consistent across groups given the same true label.
- **Equality of Opportunity** requires equal true positive rates (TPR) across groups. This ensures that each group has an equal chance of receiving a positive outcome when it is correct.
- The True Positive Rate (TPR), also known as sensitivity or recall:

  TPR= True Positives/ (True Positives + False Negatives)

$$TPR_{group\_0} \approx TPR_{group\_1}$$

$$\frac{TPR_{group\_0}}{TPR_{group\_1}} > cutoff$$

	1	0	Ratio
Race	61.1%	53.3%	0.87
Sex	63.2%	44.3%	0.70

- The ratio of TPRs across groups can be used to compare the relative differences in TPR, but it does not represent Equality of Opportunity.

Measuring Fairness – Metrics (cont.)

- **Equalized Odds** – A stronger condition than equality of opportunity. This fairness criterion is satisfied when a classifier's **True Positive Rate** (**TPR**) and **False Positive Rate** (**FPR**) are the same across groups defined by a sensitive attribute (such as race or gender).
- **Disparate Impact Ratio (**DI Ratio**)** – Disparate impact refers to a situation where a decision-making process has substantially different outcomes for different groups despite the algorithm not explicitly using sensitive attributes like race, sex, or age.
- A common threshold used to detect disparate impact, in accordance with the U.S. Equal Employment Opportunity Commission, is 0.8. Unfortunately, Canada does not have an equivalent explicit numeric threshold.

$$FPR = \frac{FP}{FP + TN}$$

$$TPR_0 = TPR_1$$
$$FPR_0 = FPR_1$$

	1	0	Ratio
Race	8.1%	3.9%	0.48
Sex	10.9%	1.7%	0.16

$$\%\ \text{predicted as positive (PPP)} = \frac{TP + FP}{N}$$

$$PPP_0 = PPP_1$$

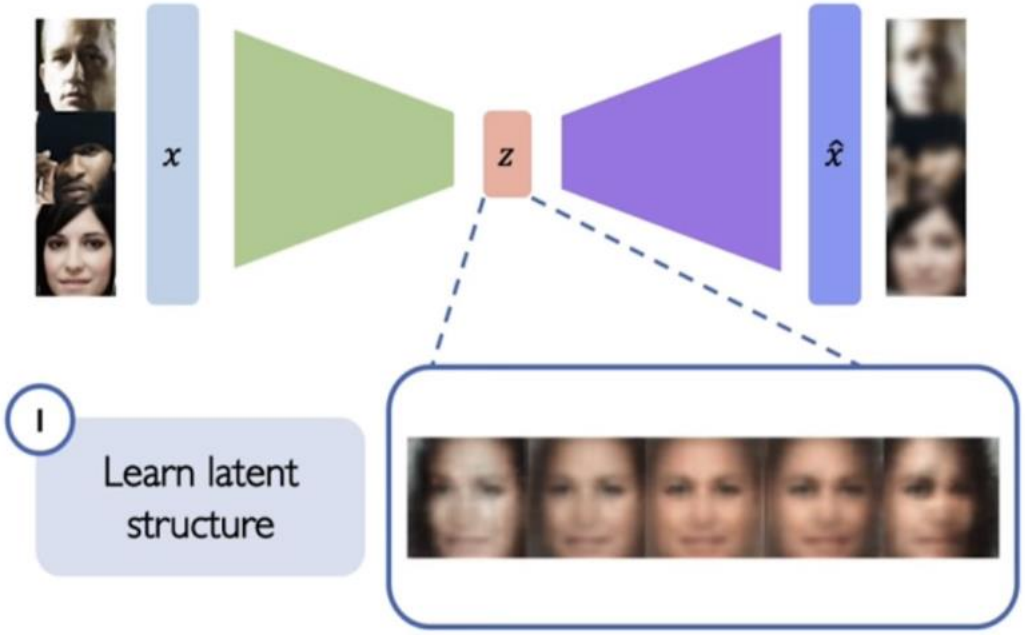
$$\frac{PPP_0}{PPP_1} > \text{Cutoff}$$

	1	0	Ratio
Race	22.0%	11.7%	0.53
Sex	27.3%	6.6%	0.24

Mitigating Biases

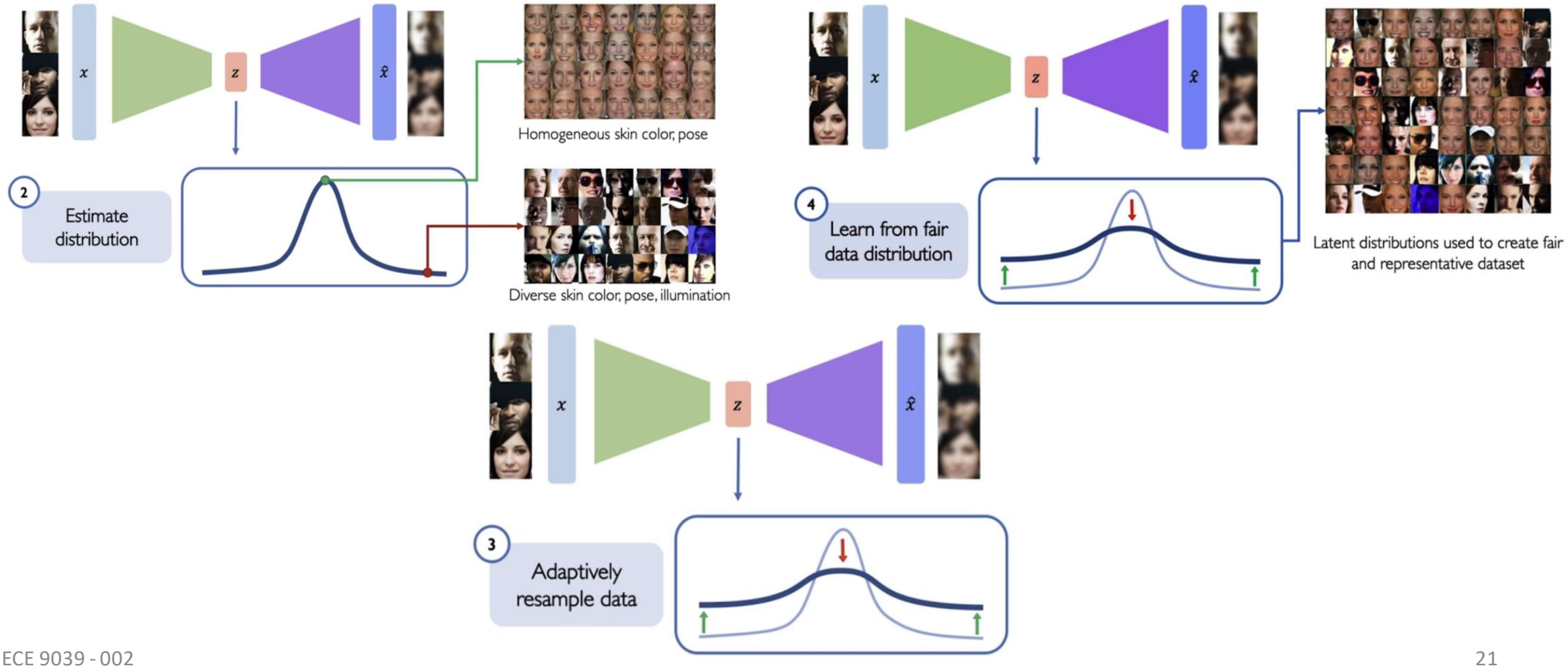
- **Balance Data**
  - Increase representation,
  - Examine and understand the data
  - Constructively engage with the issues

- **Algorithmic solutions**
  - Example: Adaptive resampling for automated debiasing
  - Uncover underlying latent variables (e.g. using Generative models)
  - Use latent distributions to identify unwanted biases



Adaptive resampling for automated debiasing

Mitigate Bias - Algorithmic Solution



References

- A. Amini, “MIT 6.S191: Evidential Deep Learning and Uncertainty”, 2021.
- A. Soleimany, “**MIT 6.S191: AI Bias and Fairness”, CMU 2021**.
- X. Bouthilier, “Practical approaches for efficient hyperparameter optimization with Oríon | SciPy”, 2021.
- R. Liaw, "A Modern Guide to Hyperparameter Optimization," 2021.
- O. Russakovsky, “Fairness in visual recognition”, Princeton University 2020.
- A. Bissuel, "Hyper-parameter optimization algorithms: a short review," 2019.
- L. Yang and A. Shami, “On hyperparameter optimization of machine learning algorithms: Theory and practice,” Neurocomputing, vol. 415, pp. 295–316, 2020.
- M. Mitchell, “cs224n-lecture19-bias”, Stanford University 2019.
- C. O'Sullivan, “Definitions of Fairness in Machine Learning”, 2023.

Attendance

You can use the provided link if you don't have a cell phone or if your phone lacks a QR-Code reader.

Federated Learning

Centralized Learning

- All data is collected and processed in a single, central location or server to train a model.
- Common Drawbacks:
  - Data Privacy and Security
  - Scalability Issues
  - Data Transfer Bottlenecks
  - Single Point of Failure

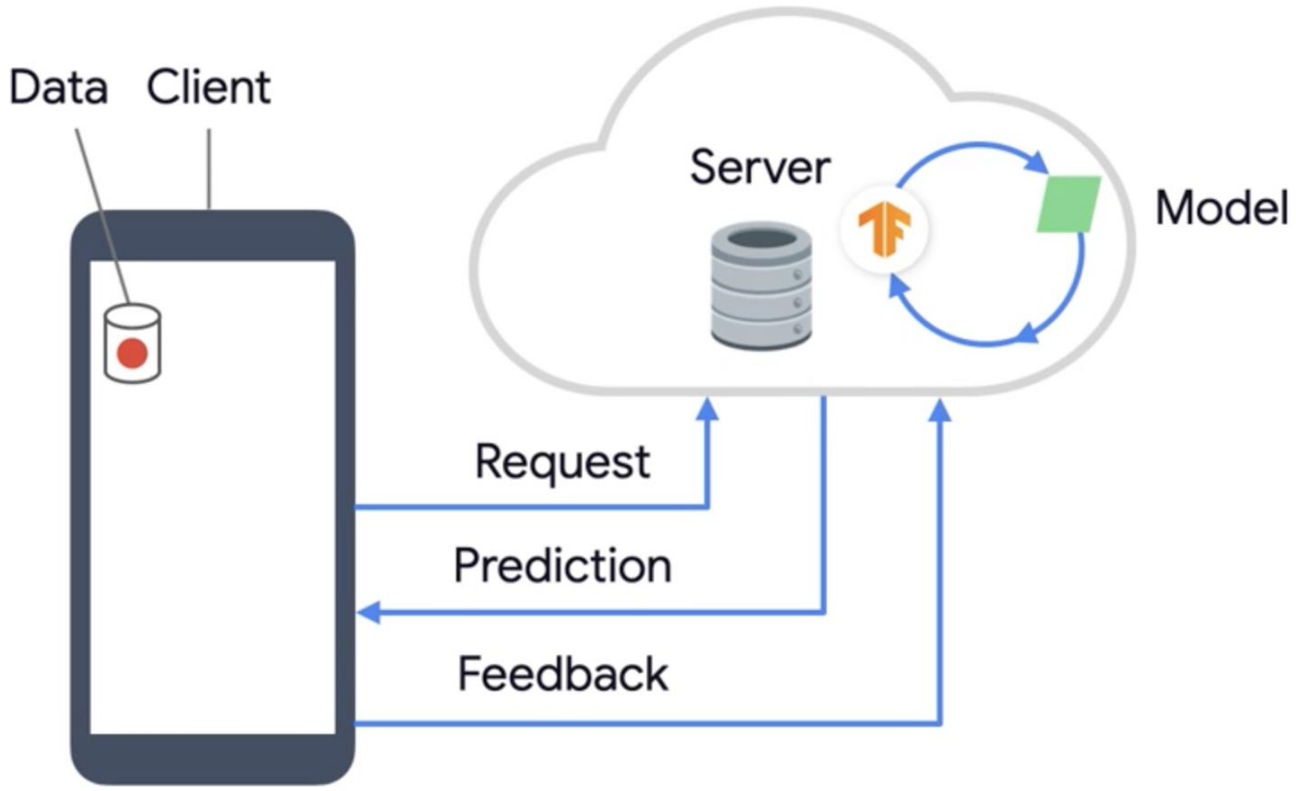
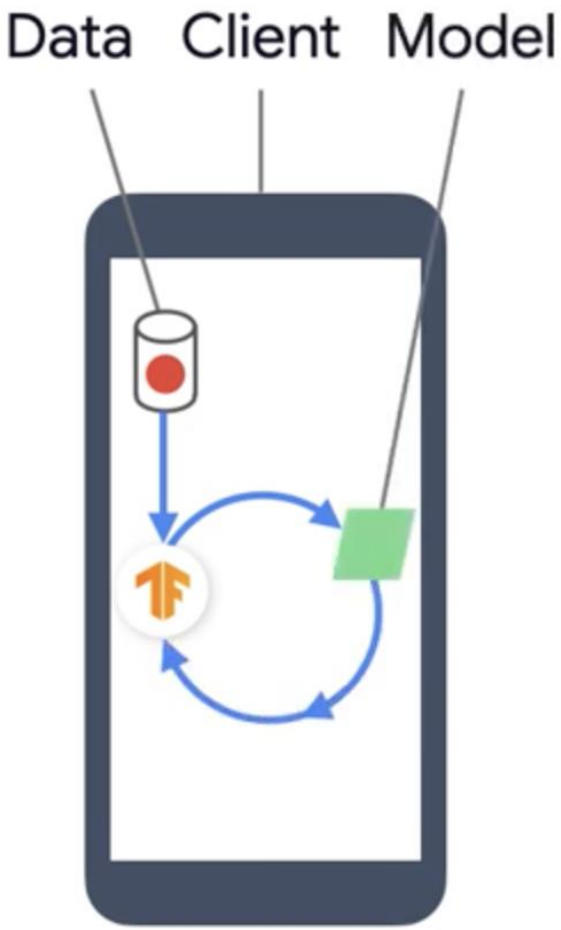


Image source:



Learning At The Edge of the Network

- This concept involves performing ML tasks directly on edge devices, such as smartphones, IoT devices, sensors, and other gadgets located at the periphery of the internet.
- Pros
  - Privacy advantages
  - Improved latency
  - Works offline
- Cons
  - Limited Resources
  - Device Heterogeneity
  - Energy Consumption
  - Quality of Service

Image source:

Pre-training Models

- This approach leverages the knowledge the model has gained during its initial training and applies it to a new but related problem.
- Training **time** is notably **reduced**.
- Pre-trained models often **perform** better than models trained from scratch.
- Utilizing pre-trained models might **save** computational resources and energy.

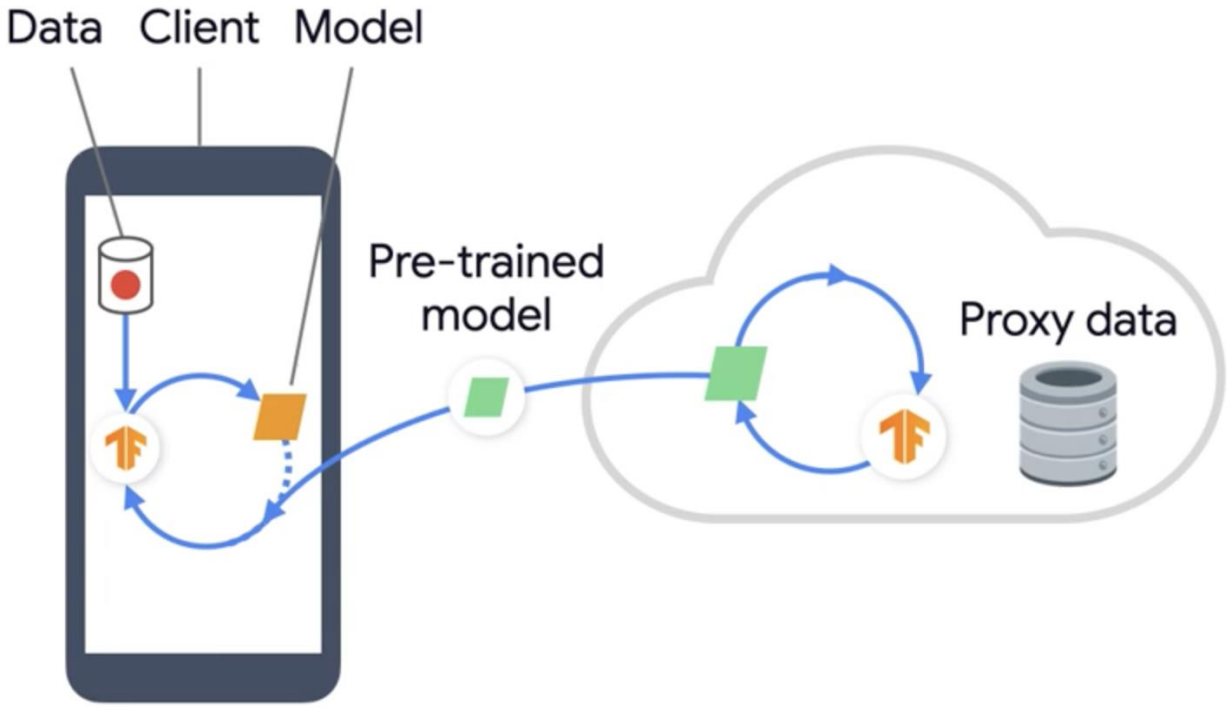


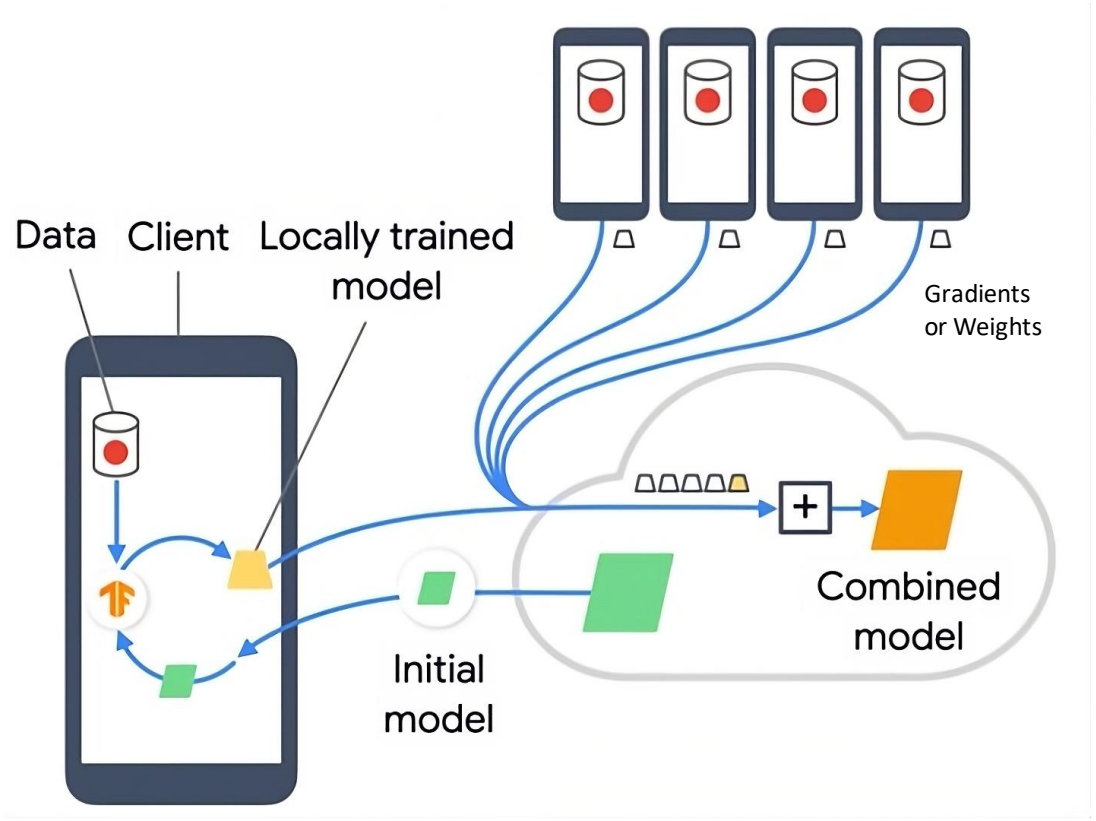
Image source:

Transfer Learning

- The learned features from one task are transferred to enhance learning on another task. This is especially effective in domains where labeled data is **insufficient** or **expensive**.

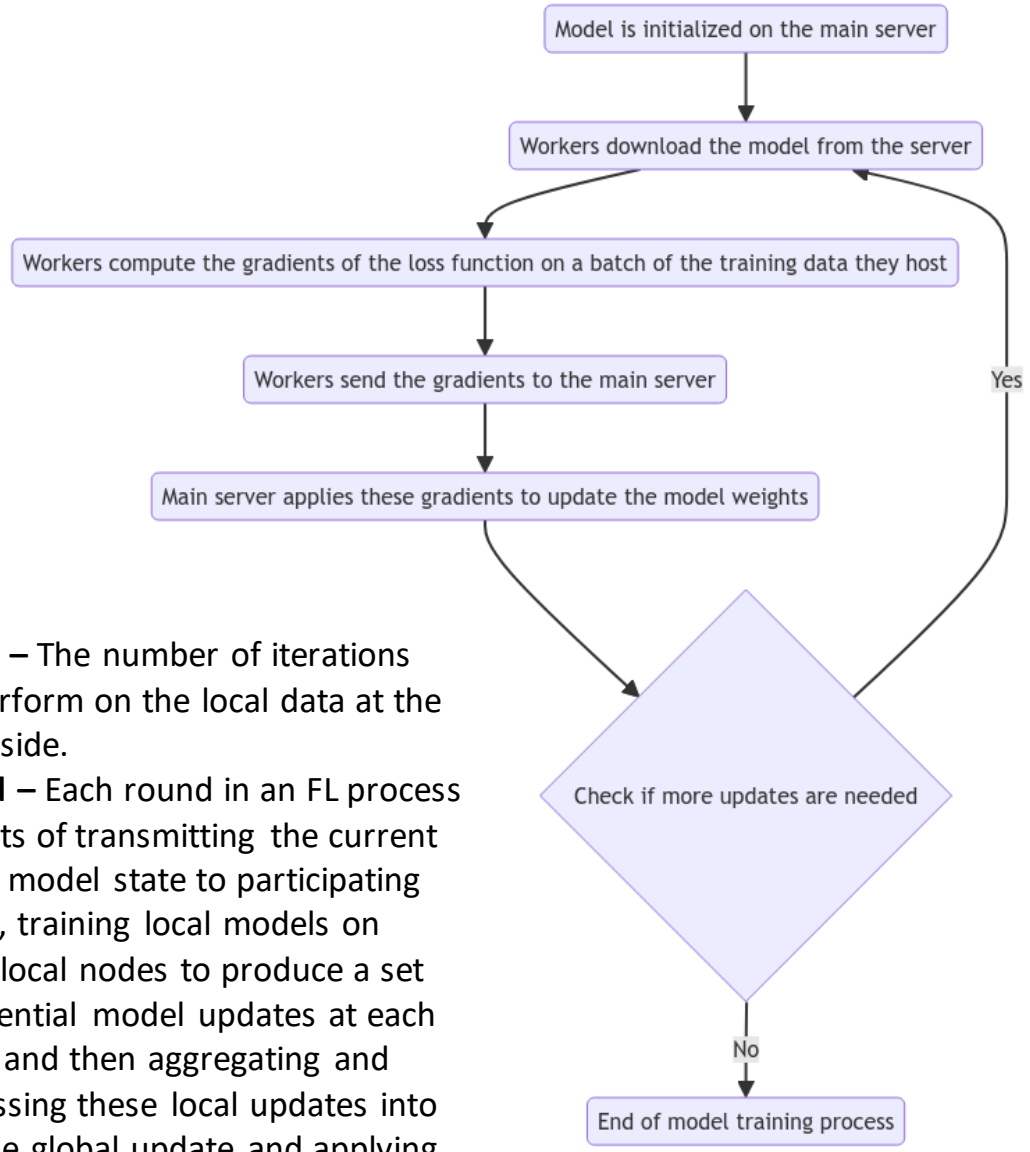
Cross-Device Federated Learning (CRFL)

- CRFL is an ML approach to train algorithms across decentralized devices holding local data, without exchanging the data.
- This method allows for the collaborative learning of a **shared model** while keeping all the training data on the device, thereby preserving privacy and security.
- CRFL used millions of **low-power devices** with **minimal data**, **reliant on low bandwidth** and **occasionally available** for training ML models collectively.



- A gradient is a vector of partial derivatives which indicates the direction in which the loss function increases fastest.

Image source:



- **Epoch –** The number of iterations we perform on the local data at the client side.
- **Round –** Each round in an FL process consists of transmitting the current global model state to participating nodes, training local models on these local nodes to produce a set of potential model updates at each node, and then aggregating and processing these local updates into a single global update and applying it to the global model.

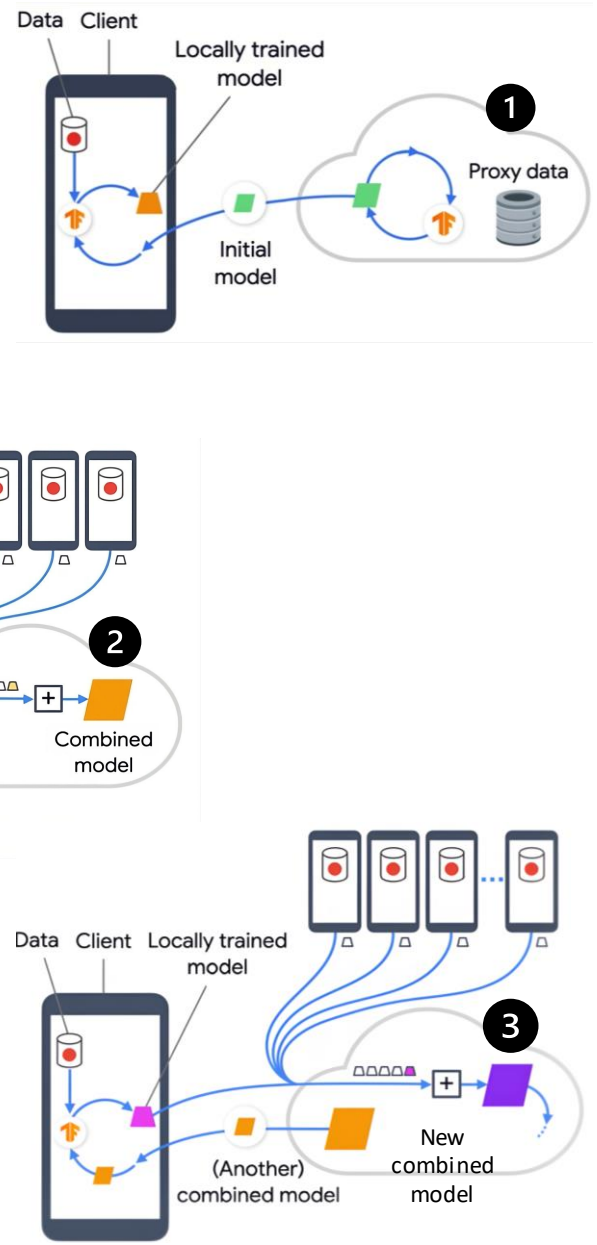
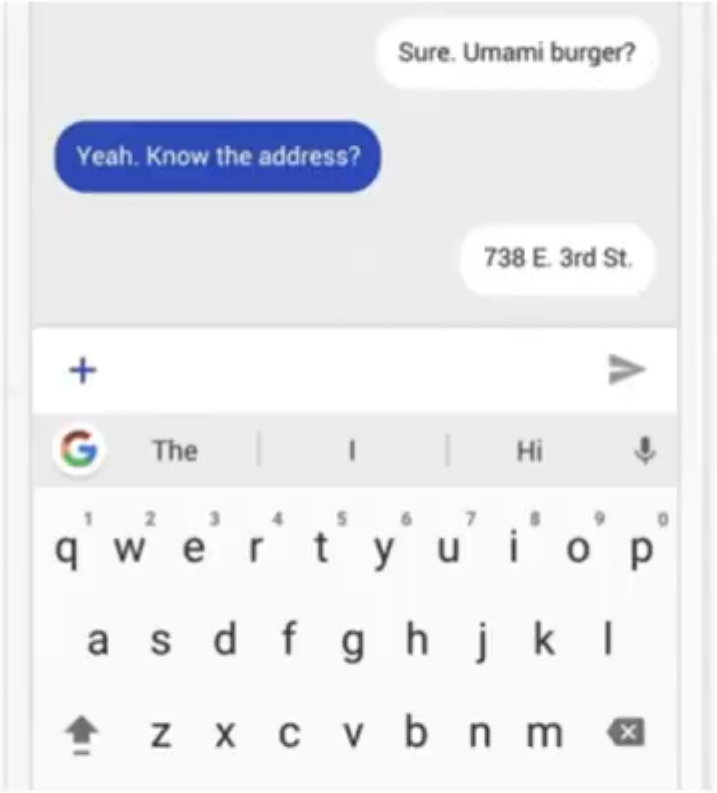


Image source:

Example: Google keyboard

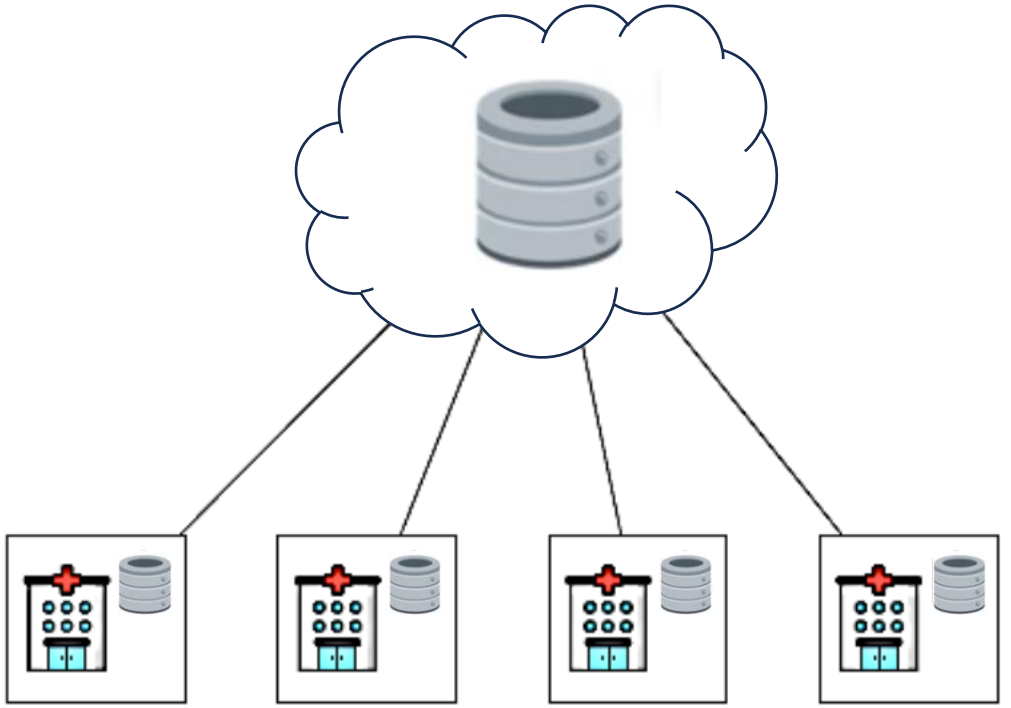
- Language modeling (Federated RNN):
  - Better next-word prediction
  - Emoji prediction – more accurate emoji predictions
  - Action prediction – When is it useful to suggest a gif, sticker, or search query?: Reduction in unhelpful suggestions



source:

Cross-Silo Federated Learning (CSFL)

- CSFL is another type of federated learning that occurs across different organizations or "**silos**" rather than individual user devices.
- The term "**silo**" implies that each entity's data is stored separately due to privacy concerns, regulatory requirements, or business competition.
- Cross-silo federated learning connects tens to hundreds of **high-power devices (servers)** with **sufficient data**, **utilizing high-bandwidth links** to ensure constant availability for robust model training.



Challenges & Constraints of Federated Learning

Constraints in the Federated Setting

- Devices must be **idle**, **charged**, and **connected** to Wi-Fi to participate in the learning process.
- Availability for federated learning depends on geographic location.
- Implementing new device operations requires advance planning due to dependency on client software release cycles.
- Data from devices is inherently noisy; thus, models **must** be robust and clean.
- Server-side **data normalization** routines are too huge for devices and must be adapted and minimized for on-device use.
- Hyperparameter optimization on the client side, like batch size and learning rate adjustments, is necessary for rapid convergence and may include techniques such as **gradient clipping**.
- Training is organized into **rounds**; minimizing the number and duration of these rounds is beneficial, with transfer learning techniques enhancing convergence speed.

Data Availability & Heterogeneity

Available

Not available

- **Data Availability** is key as federated learning relies on accessing local data from many devices to update and improve a central model.
- Further, data is distributed across many devices, leading to models that perform well on some nodes but **poorly** on others, complicating the training process and affecting overall model performance.

source:

System Heterogeneity & Communication Overhead

- The devices participating in federated learning can vary widely in their **computational power**, **memory**, **storage**, and **network connectivity**. This can lead to differences in training times, with some devices becoming **bottlenecks** for the entire learning process.
- Federated learning requires regular communication between the central server and the devices to send model updates and aggregate them. These communication costs can become unreasonable in environments with limited or costly bandwidth.

Available

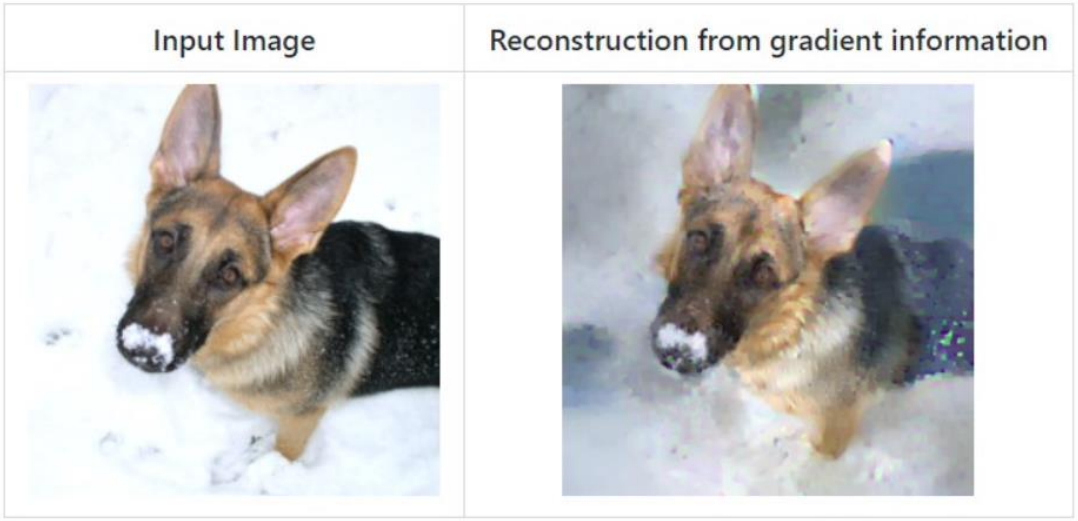
Suitable available

source:

Improving Privacy in Federated Learning

Privacy Concerns

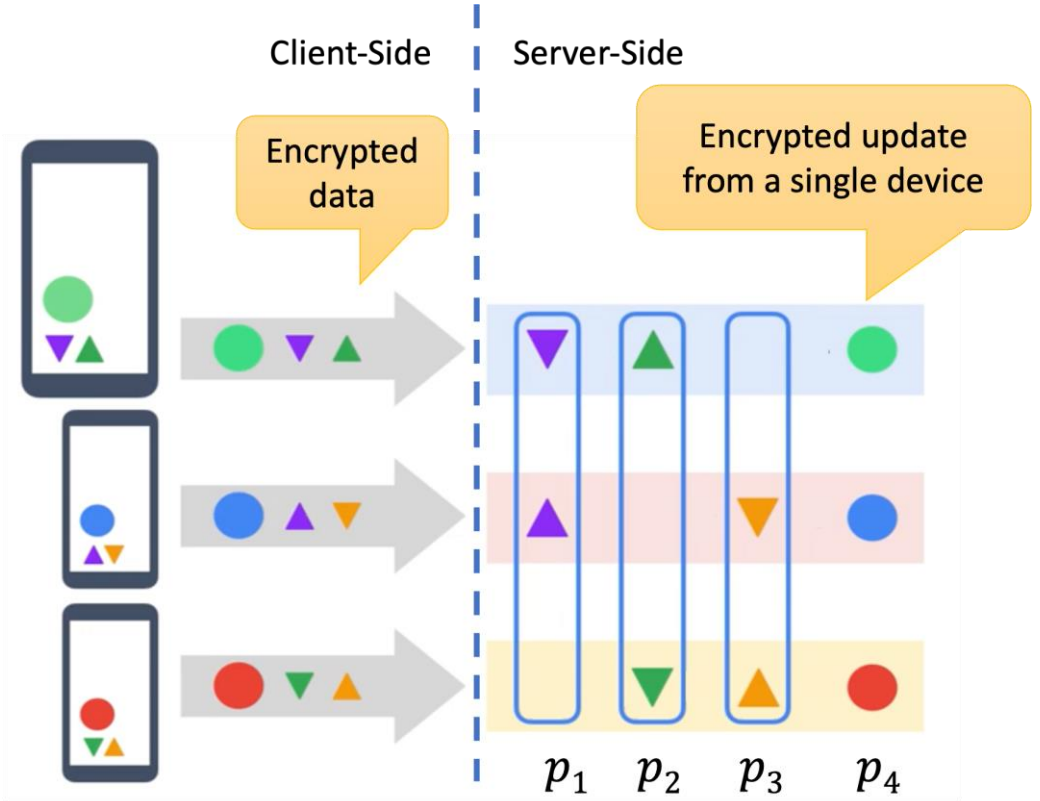
- Gradient inversion attacks (reconstruct the original training data from the shared gradients).
- Model updates leak unintended information about clients' training data.
- Solutions
  - **Secure Aggregation** of model updates from multiple clients
  - Train using **Differential Privacy**



J. Geiping, et al. Inverting gradients-how easy is it to break privacy in federated learning?. Advances in Neural Information Processing Systems, 33, pp. 16937-16947, 2020.

Secure Aggregation (SG)

- In a federated learning scenario, multiple devices train a model on their local data and then send only the model updates (such as **gradients** or **weights**) to a central server.
- Each device encrypts its model update using cryptographic techniques before sending it to the central server.
- Pairs cancel out when aggregated together, revealing only the sum.

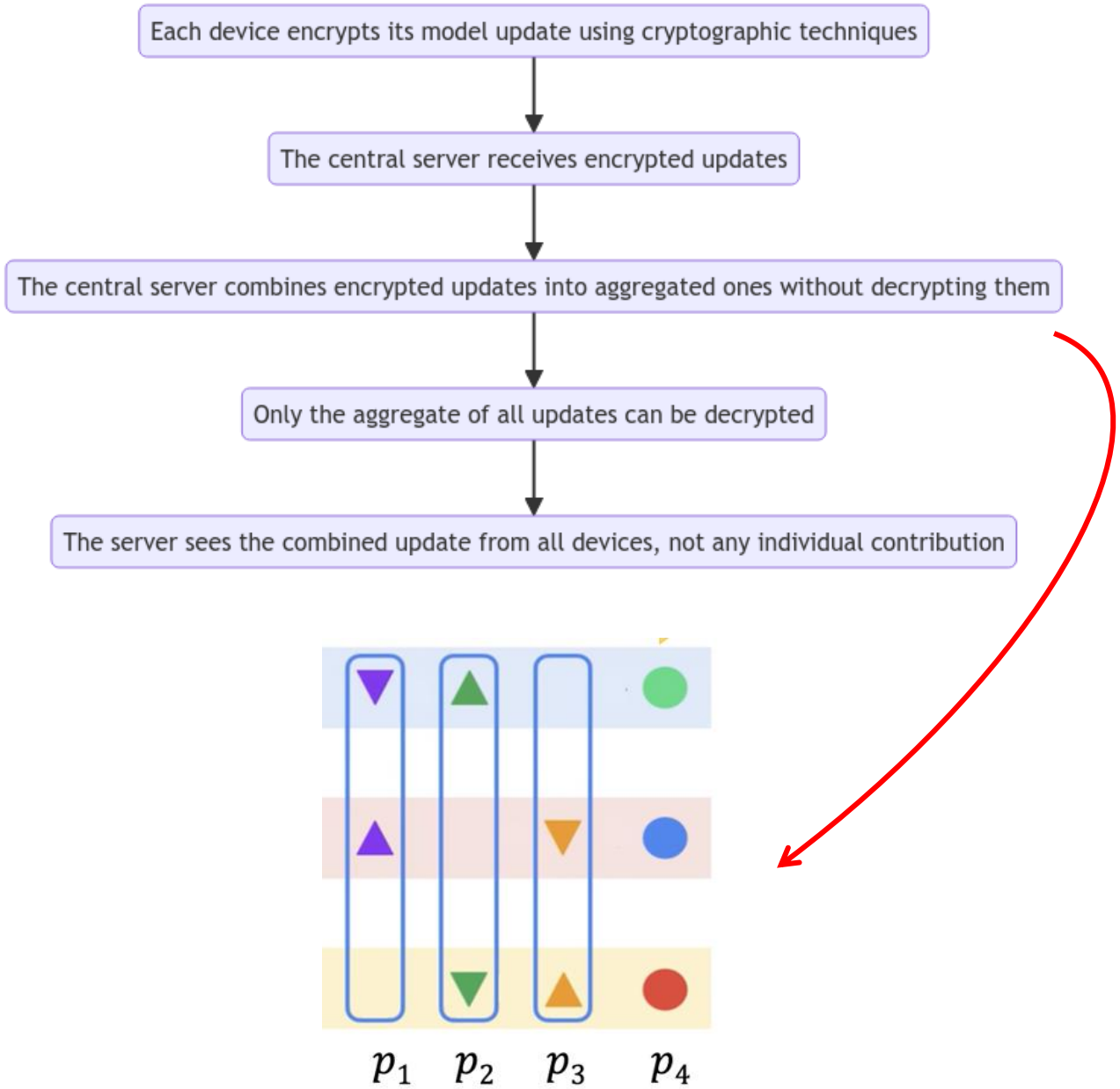


Visualizing how multiple encrypted inputs (updates from various devices) are combined to produce an aggregate sum (*Vector*)

source:

How does SG apply?

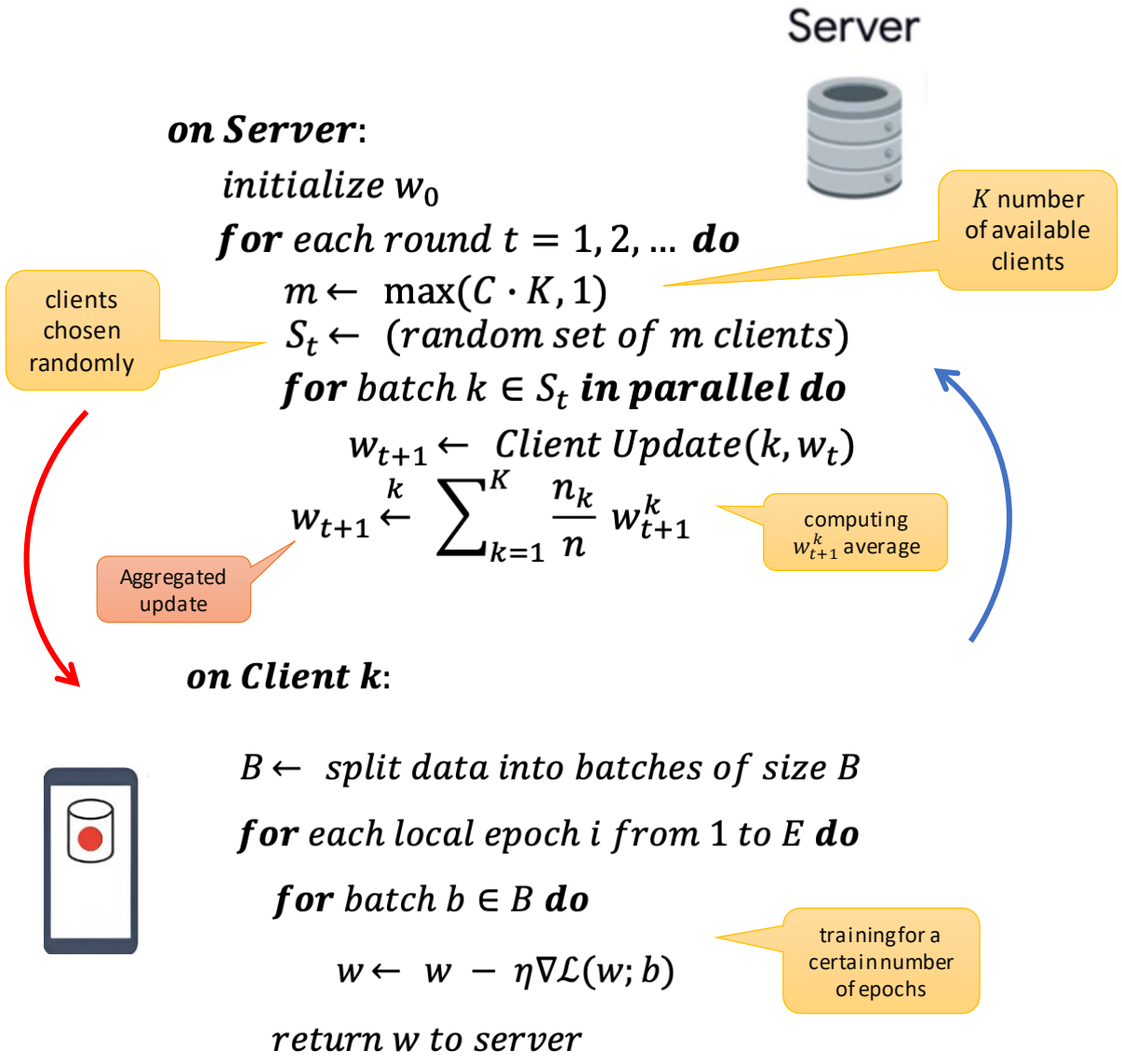
- **Encryption**: Each device encrypts its model update using cryptographic techniques before sending it to the central server.
- **Aggregation**: The central server receives encrypted updates and combines them into an aggregated update (*vector*). The server does this without being able to decrypt and see any individual update.
- **Decryption**: Only the aggregate of all updates can be decrypted, meaning the server only sees the combined update from all devices and not any individual contribution.



source:

Federated Averaging / Local SGD

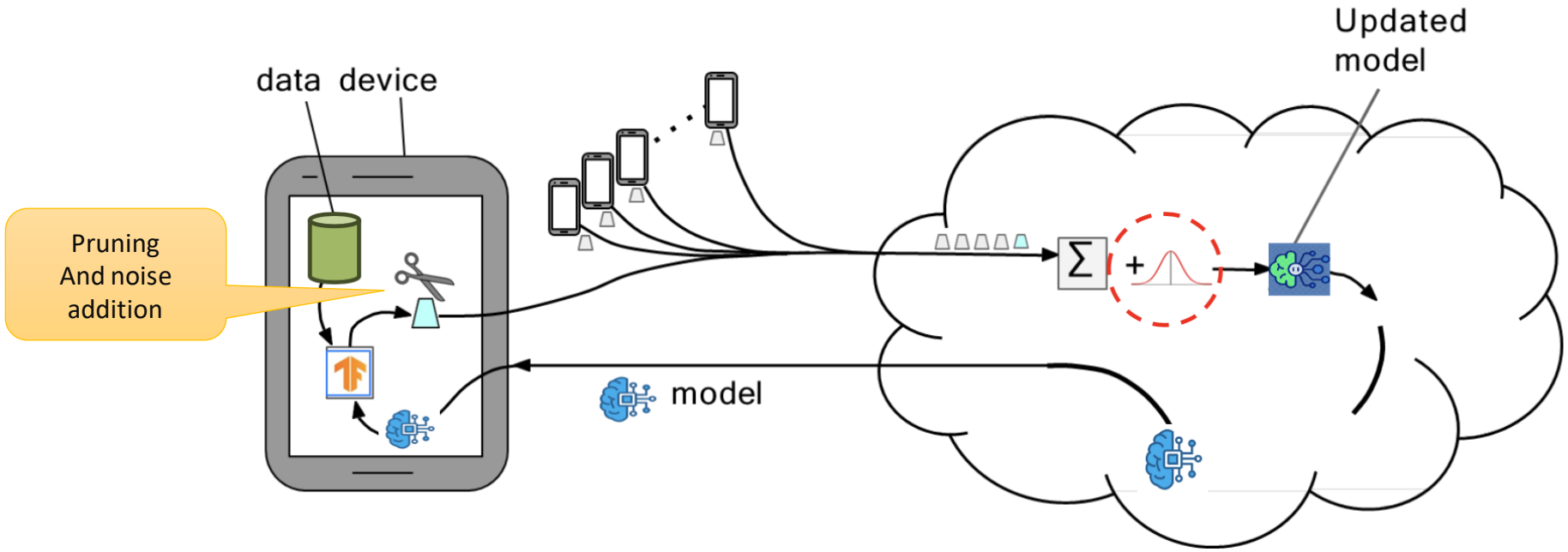
- Federated Averaging (*FedAvg*) is an algorithm in federated learning that trains a global ML model using decentralized data residing on **edge devices**.
- The **aggregated update** is intended to approximate the gradient that would have been computed if all the data were centralized.



source: Google

Differential Privacy

- Learning common patterns in a dataset without memorizing individual examples.
- Use noise to obscure an individual's impact on the learned model.



H. B. McMahan, et al. Learning Differentially Private Recurrent Language Models. ICLR 2019.

References

- A. Amini, “MIT Introduction to Deep Learning”, 2023.
- B. McMahan, “Federated Learning, from Research to Practice”, CMU 2019.
- F. Beaufays, “Federated learning: Basics and application to the mobile keyboard”, ML Tech Talks 2021.
- M. Rabbat, “Federated Learning at Scale” Meta AI”, 2022.
- A. Bellet, "Introduction to Federated learning," 2020

Generative Models

Discriminative vs. Generative models

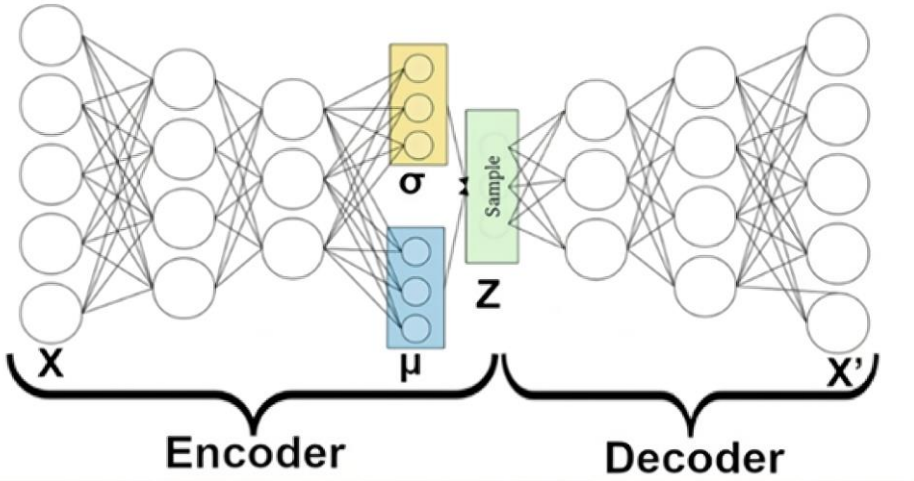
- **Discriminative models** learn the boundary between classes or predict the probability of a label given the data features. They focus on differentiating between different types of data points.
  - Ex.: Logistic regression, SVM, and most neural networks are discriminative models
- **Generative models** refer to algorithms designed to generate new content, ideas, or data that were not explicitly input into the system. These AI systems can produce text, images, music, code, and other forms of creative output that mimic human-like creativity and innovation.

Examples of Generative Models

- **Large Language Models (LLMs)** like GPT (Generative Pre-trained Transformer) are a subset of Generative AI and typically use the Transformer architecture.
- **Variational Autoencoders (VAEs)** generate complex data like images and music by learning a compressed representation of the input data and then developing new data from this compressed representation.

Recall: Variational Autoencoder (VAE)

- Instead of directly converting input data to specific points in the latent space, VAE converts them to parameters that define a **probability distribution**.
- This approach dictates where a data point will likely be positioned in the latent space based on its characteristics.
- The VAE encoder outputs a probability distribution for **each latent attribute**.
- So, How does this help?
  - The VAE learns to rebuild not just from specific encoded points, but also from their surrounding space, enabling it to create new data by sampling from regions in that space rather than just replicating existing data tied to fixed points.



Source: Chouikhi, F., Abbes, A.B., Farah, I.R. (2023). Desertification Detection in Satellite Images Using Siamese Variational Autoencoder with Transfer Learning. In: Nguyen, N.T., et al. Computational Collective Intelligence. ICCCI 2023. Lecture Notes in Computer Science(), vol 14162. Springer, Cham.

$$Z = \mu + \sigma \odot \epsilon$$
$$\epsilon \sim \mathcal{N}(0, 1)$$

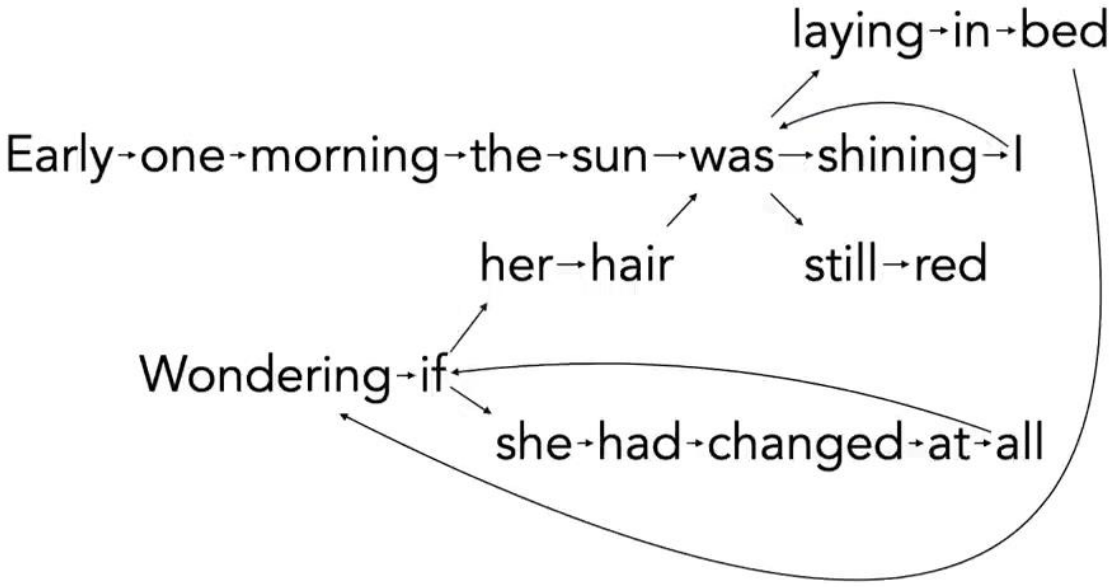
A Language Model

Early one morning the sun was shining I was laying in bed

Wondering if she had changed at all if her hair was still red

Source: https://www.youtube.com/watch?v=lnA9DMvHtfI

A Language Model (Cont.)

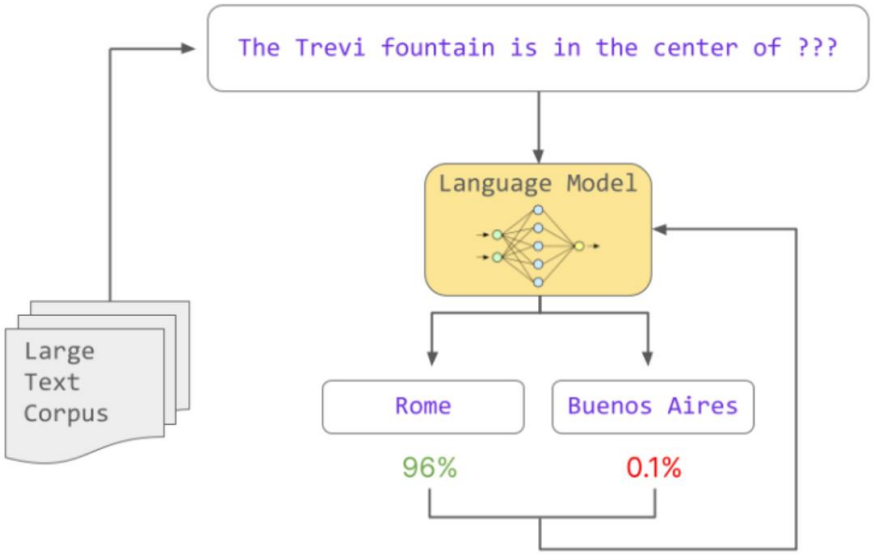


Source: https://www.youtube.com/watch?v=lnA9DMvHtfI

"Large" Language Models

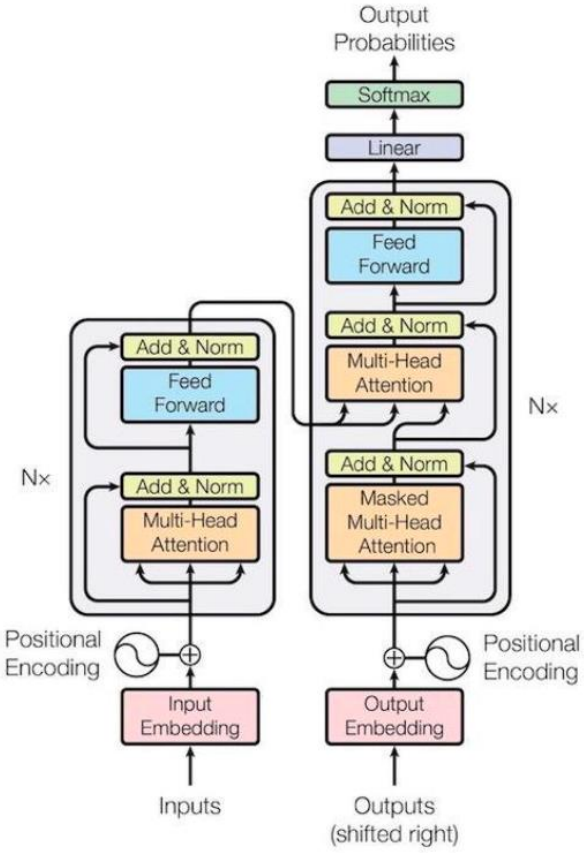
- Language Models (LMs) are probabilistic models explicitly tailored to identify and learn statistical patterns in natural language.
- Why "Large"?
- LLMs undergo a **training phase,** during which they learn from a **large** text corpus. This process involves adjusting internal parameters to accurately predict the next word in a sequence based on the context provided by preceding words.
- During the **inference phase**, the trained model uses the patterns it learned to generate text or complete prompts without needing to access the large text corpus directly.

Prediction Phase
(Inference):



Breakthrough in LM: "Attention is all you need"

- Transformer model architecture introduced in 2017:
  - **Encoder**:
    - Assign to each unique word a unique identifier (a number serves as a **token** to represent that word).
    - Note the location of every token relative to every other token.
    - Using just **token** and **location**—determine the probability of it being adjacent to, or in the locality of, every other word.
  - Feed these probabilities into a NN to build a map of relationships.
  - Given any string of words, NN predicts the next word (e.g., **AutoCorrect**)
  - **Decoder** – Takes in token representation and decodes it back into text.
- Depending on the task, a language model may use only the encoder part (e.g., BERT), only the decoder part (e.g., GPT), or both (e.g., M2M-100).



Bidirectional Encoder Representations from Transformers (BERT)

- It utilizes only the encoder part of the Transformer architecture. The encoder reads and processes the entire input sequence at once.
- BERT's encoders are designed to simultaneously capture context from both directions (left and right of each word in the sequence). This is known as bidirectional context, which is powerful for understanding the meaning of words in context.
- Encoder-only models are particularly good at tasks that require understanding the context around each word, such as sentence classification and question answering, where the answer is contained within the provided context.

Generative Pre-trained Transformer (GPT)

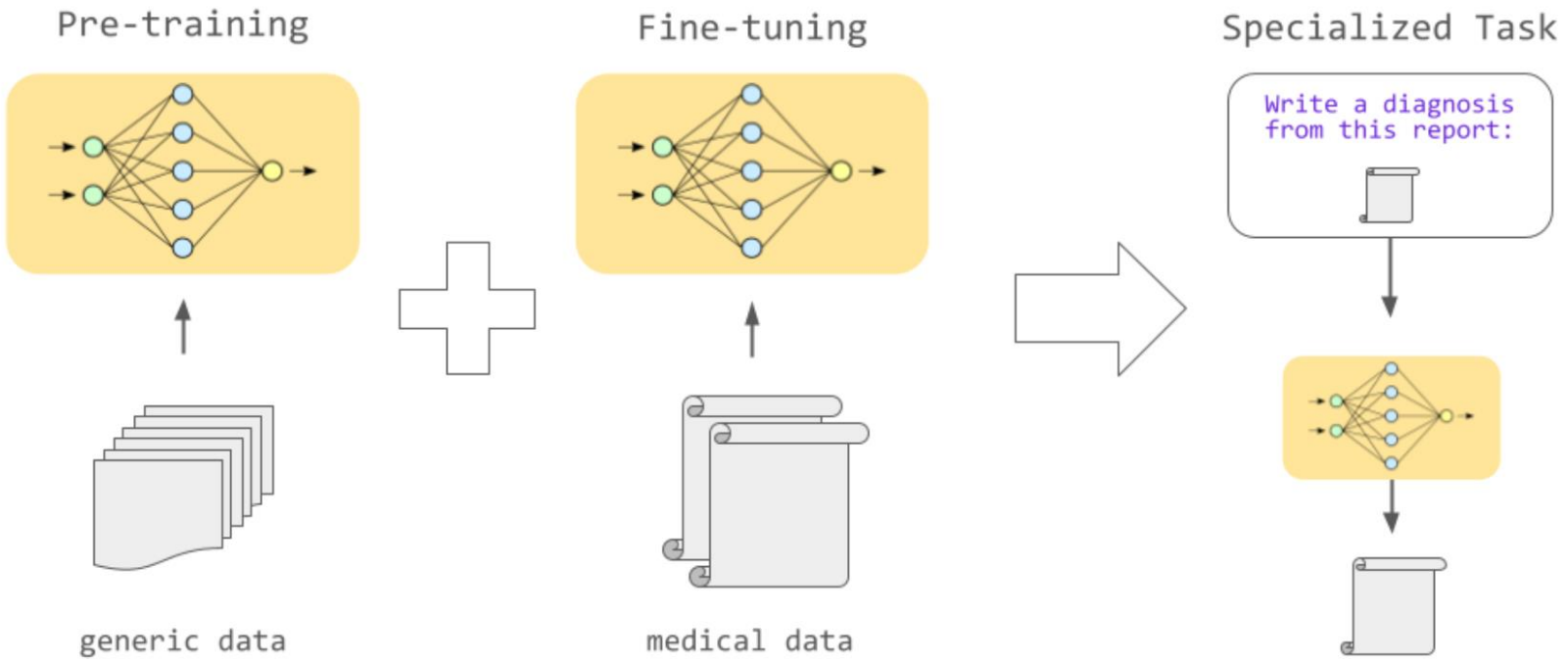
- GPT Employs only the decoder part of the Transformer architecture to understand the input and generate the output.
- The decoder generates an output sequence, one piece at a time, from left to right. It utilizes a **unidirectional context**, meaning each token can only attend to previous tokens, not future ones.
- Decoder-only models are typically used for **generative tasks** like text completion, storytelling, and anywhere else where the goal is to produce new content based on a prompt.

Multilingual Machine Translation (M2M-100)

- M2M-100 uses both the encoder and decoder components.
- The **encoder** processes the input sequence to understand its context, and the **decoder** uses this understanding to generate an output sequence. This setup allows the input and output sequences to be in different languages or formats.
- Encoder-decoder models are ideal for tasks that involve converting an input sequence to an output sequence where the sequences are related but not identical, such as in **machine trans**lation, **summarization**, and **speech recognition**.

Fine-Tuning LLM & Transfer Learning

- Adapt a pre-trained model to technical or specialized knowledge domains, such as the medical or legal fields



Source: M. Ramponi, "The Full Story of Large Language Models and RLHF", 2023

Future of Generative Models

- Generative models are used in many applications (drug design, material science, chip design, synthetic data, design and manufacturing, business planning, etc.)
- By 2025, 30% of outbound marketing messages from large organizations will be synthetically generated, up from less than 2% in 2022.
- By 2030, a major blockbuster film will be released with 90% of the film generated by AI (from text to video), from 0% of such in 2022.

Source: https://www.sequoiacap.com/article/generative-ai-a-creative-new-world/

For attending
ECE 9039