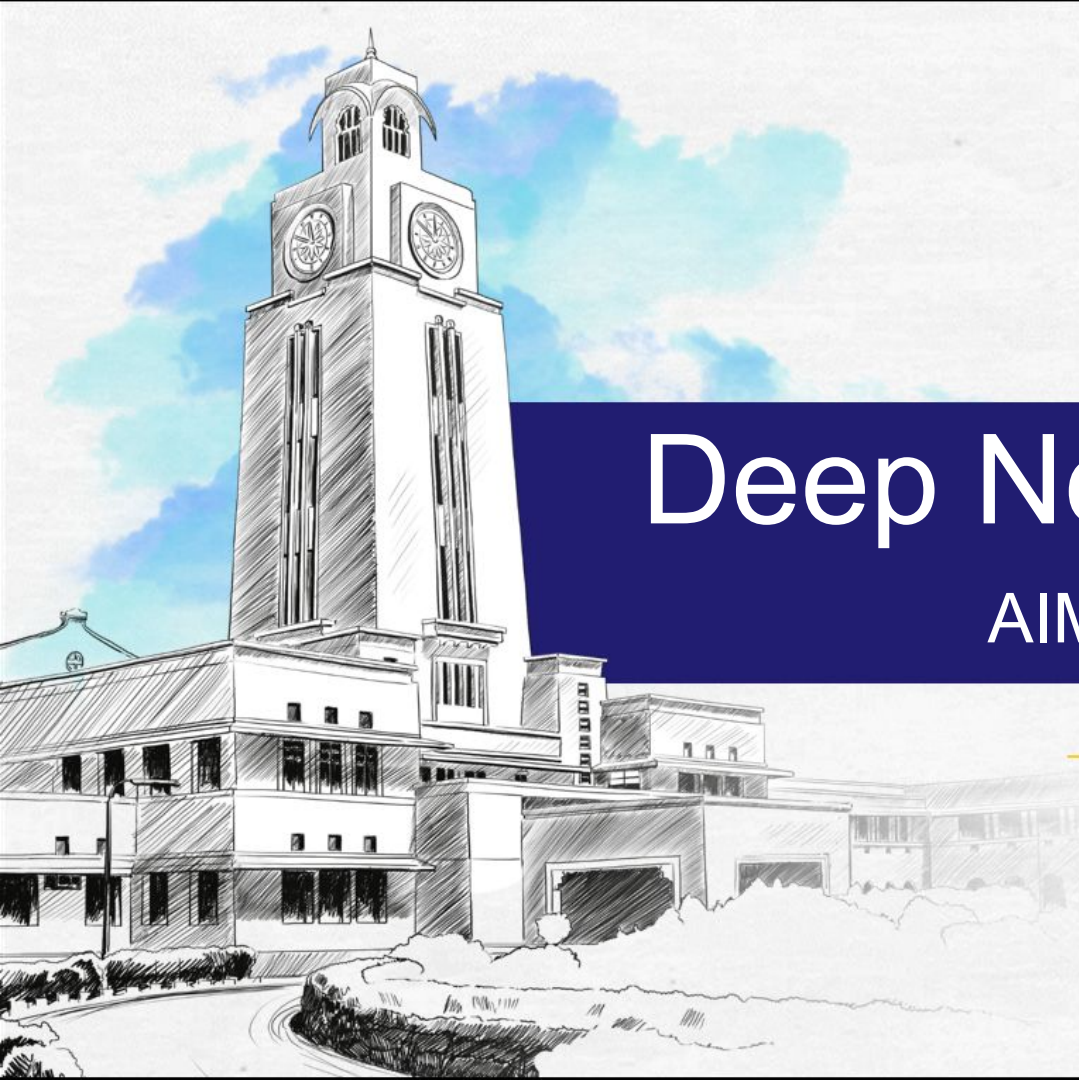


BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Deep Neural Network

AIML Module 10

Seetha Parameswaran

BITS Pilani

The author of this deck, Prof. Seetha Parameswaran, is gratefully acknowledging the authors who made their course materials freely available online.

What we Learn….

10.1 Federated learning

10.2 Meta learning

10.3 Online (incremental) learning

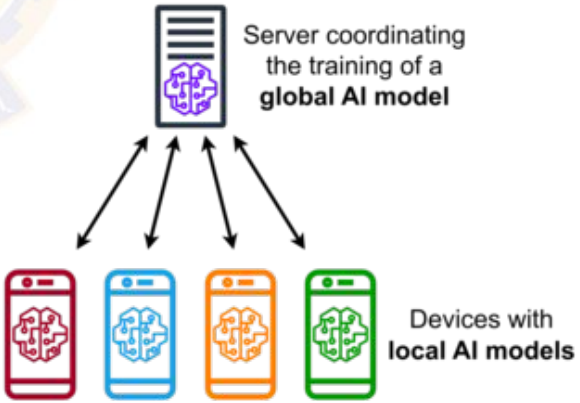
Federated Learning

Federated Learning

- Federated learning is a way to train AI models without anyone seeing or touching your data.
- Google introduced the term federated learning in 2016.
- AI models are being trained on the edge, on data that never leave the source of the data like laptop, server or mobile phone or wearable device.
- Decentralized approach.

Federated Learning

- Federated Learning is a machine learning approach that allows a model to be trained across multiple decentralized edge devices (such as smartphones, IoT devices, or local servers) holding local data samples, without exchanging them centrally.
- Instead of sending all data to a central server for model training, federated learning keeps data localized, thus addressing privacy, security, and bandwidth concerns.



Federated Learning

- Read world performance of ML model depends on **relevance** of the data used to train it.
- The best data is available at the **source** , ie the devices used **every day**.

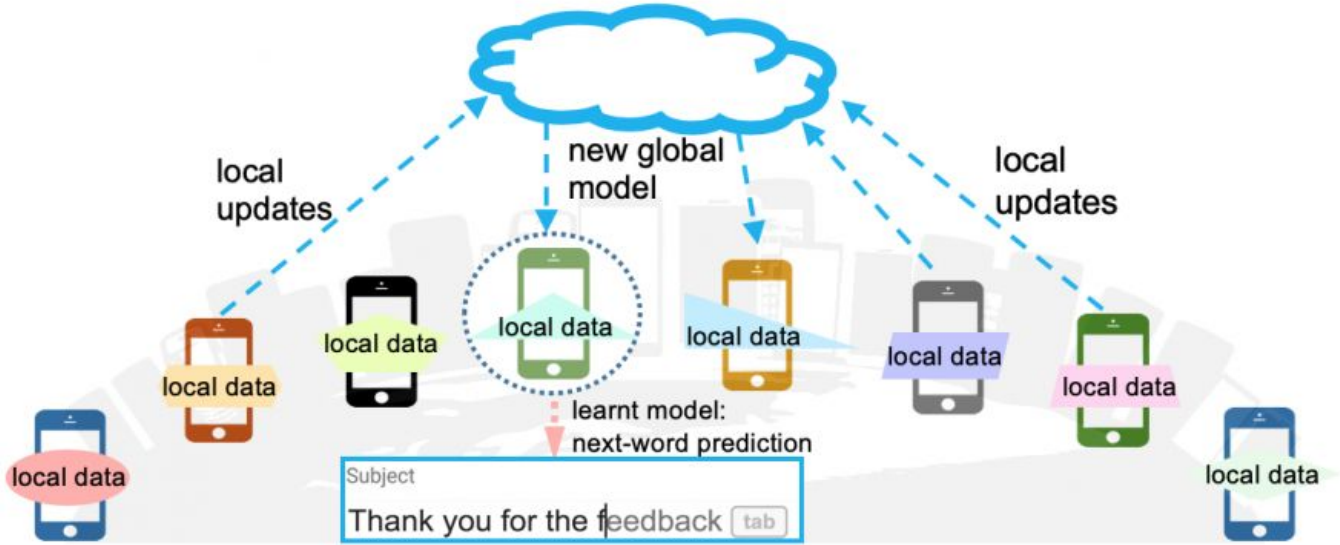
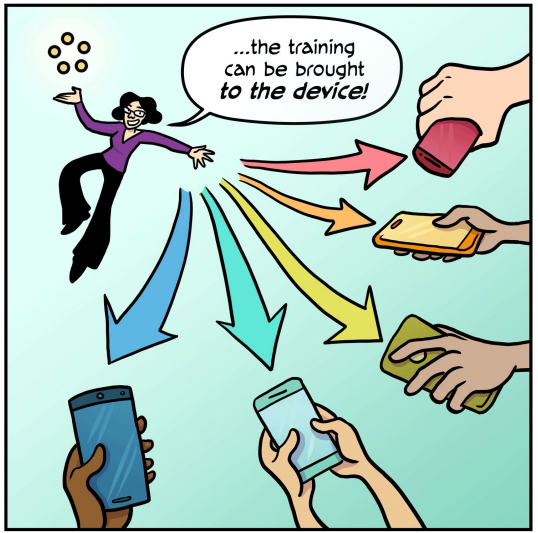


Image: https://blog.ml.cmu.edu/2019/11/12/federated-learning-challenges-methods-and-future-directions/

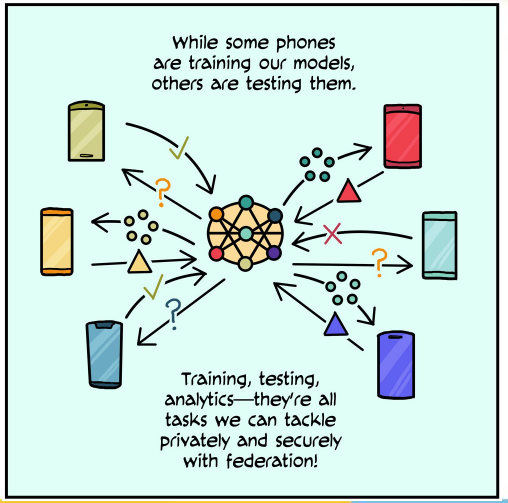
Federated Learning

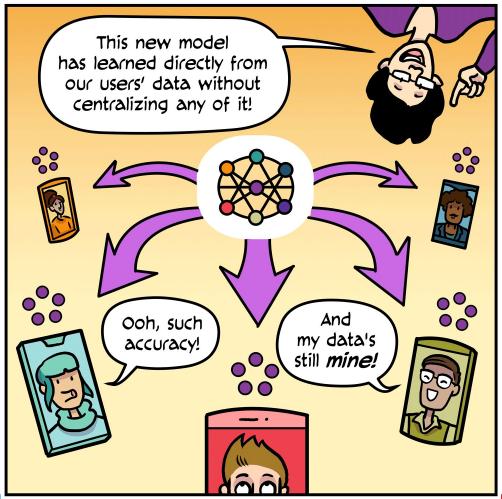
- Federated Learning = train a centralized model on decentralized data.
  - On-device data is used to train a central model. Data resides on the device and is not brought to the server on which training is performed.
  - Training is brought to the device. (Some challenges: Battery, wifi, idle, compute)
  - Devices participate when they are eligible. A subset of devices are selected to receive a training model. The model may be few megs.



Federated Learning

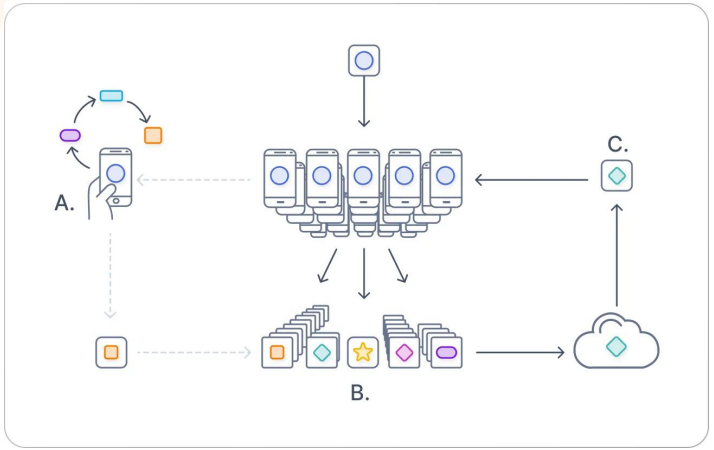
- Federated Learning = train a centralized model on decentralized data.
  - Training on the data on the device in few minutes, sends the training results to the server. The results are encrypted with a key that the server does not have.
  - Uses **Secure aggregation and Differential privacy**
  - ML models are trained on device after multiple rounds of training.
  - Test the ML model on the user devices.
  - New model is **static**. It learned from thousands of users who may have rare unique data.





Types of Federated Learning

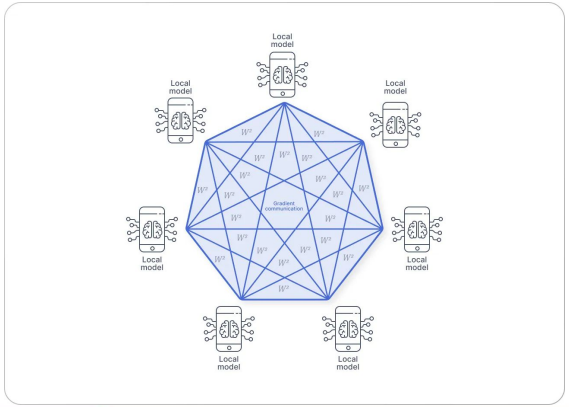
1. Centralized federated learning
   - Centralized federated learning requires a central server.
   - It coordinates the selection of client devices in the beginning and gathers the model updates during training.
   - The communication happens only between the central server and individual edge devices.
   - Straightforward approach
   - Generates accurate models
   - Central server poses a bottleneck problem



https://ai.googleblog.com/2017/04/federated-learning-collaborative.html

Types of Federated Learning

2. Decentralized federated learning
   - Decentralized federated learning does not require a central server to coordinate the learning.
   - The model updates are shared only among the interconnected edge devices.
   - The final model is obtained on an edge device by aggregating the local updates of the connected edge devices.
   - Prevents the possibility of a single-point failure.
   - The model's accuracy is completely dependent on the network topology of the edge devices.



https://www.researchgate.net/figure/Decentralized-Federated-Learning-Environments_fig1_344386575

Types of Federated Learning

3. Heterogeneous federated learning
   - Heterogeneous federated learning involves having heterogeneous clients such as mobile phones, computers, or IoT (Internet of Things) devices. These devices may differ in terms of hardware, software, computation capabilities, and data types.
   - HeteroFL can generate a single global model for inference after training over multiple varied local models.

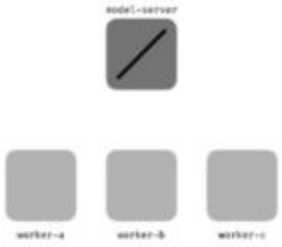
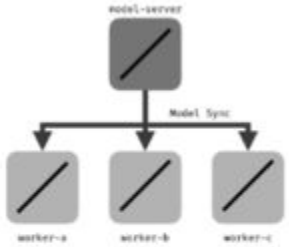
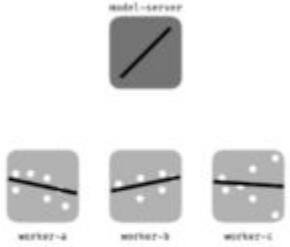
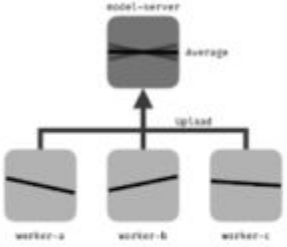
Federated Learning Algorithms

1. Federated stochastic gradient descent (FedSGD)
   - In FedSGD, the central model is distributed to the clients, and each client computes the gradients using local data. These gradients are then passed to the central server, which aggregates the gradients in proportion to the number of samples present on each client to calculate the gradient descent step.
2. Federated averaging (FedAvg)
   - Clients can perform more than one local gradient descent update. Instead of sharing the gradients with the central server, weights tuned on the local model are shared. Finally, the server aggregates the clients' weights (model parameters).
   - If all the clients begin from the same initialization, averaging the gradients is equal to averaging the weights. Therefore, Federated Averaging leaves room for tuning the local weights before sending them to the central server for averaging.
3. Federated learning with dynamic regularization (FedDyn)
   - In federated learning, the global loss must be computed based on local losses generated from heterogeneous devices.
   - Due to the heterogeneity of clients, minimizing global loss is different than minimizing local losses. Therefore, FedDyn method aims to generate the regularization term for local losses by adapting to the data statistics, such as the amount of data or communication cost. This modification of local losses through dynamic regularization enables local losses to converge to the global loss.

Federated Learning

- Secure aggregation enables server to combine the encrypted results and only decrypt the aggregate.
  - On each device, before anything is sent, the secure aggregation protocol adds zero-sum masks to scramble training results.
  - Add up all the training results to cancel out the masks.
- Unique rare data.
  - ML model should not memorize data specific to a device.
  - Add noise to obscure rare data. This is known as differential privacy.
- Differential privacy
  - Deal with the risk of model memorization, where a shared model's parameters may be influenced heavily by a single data.
- Federated Learning = ML model learns from everyone without learning about anyone.

Steps in Federated Learning

Step 1	Step 2	Step 3	Step 4
			
Central server chooses a statistical model to be trained	Central server transmits the initial model to several nodes	Nodes train the model locally with their own data	Central server pools model results and generate one global mode without accessing any data

Steps in Federated Learning

1. Initialization:
   a. A global model is initialized on a central server. This model serves as a starting point for training.
   b. A ML model (e.g., linear regression, neural network, boosting) is chosen to be trained on local nodes and initialized.
2. Client selection
   a. A fraction of local nodes are selected to start training on local data.
   b. The selected nodes acquire the current statistical model while the others wait for the next federated round.
3. Local Training:
   a. Each edge device performs local model updates using its own local data but does not send data samples to the central server.
   b. The local model is trained on the device using a portion of the data.
   c. The device computes the gradients of the local model with respect to its data but only shares these gradients with the central server, not the data itself.
4. Aggregation:
   a. The central server collects gradient updates from all participating devices.
   b. It aggregates these updates to update the global model.
   c. Popular aggregation methods include federated averaging, weighted averaging, or other secure and privacy-preserving aggregation techniques.
5. Global Model Update:
   a. The updated global model is sent back to the participating devices.
6. Local Model Fine-Tuning:
   a. Devices further fine-tune the global model locally using their data to adapt it to their specific characteristics or changes in data distribution.
7. Repeat:
   a. Steps 2 to 6 are repeated iteratively for several rounds, allowing the global model to improve based on local data without centralizing it.

Federated Learning - Advantages

- Privacy:
  - User data remains on their devices, reducing privacy concerns associated with centralized data storage and processing.
- Efficiency:
  - Bandwidth and computational resources are saved because only model updates (gradients) are communicated, not raw data.
- Security:
  - The decentralized nature of federated learning makes it more resilient to certain attacks, such as data breaches during centralized data aggregation.
- Personalization:
  - Local fine-tuning allows the global model to adapt to individual device characteristics or data distributions, enabling personalized recommendations or predictions.

Federated Learning

- Use cases for federated learning
  - Transportation: self-driving cars
    - Training self-driving cars on aggregated real-word driver behaviour.
  - Medicine: digital health
    - Help hospitals improve diagnostics while maintaining patient privacy.
  - Industry 4.0: smart manufacturing
    - Improve the efficiency and effectiveness of industrial process while guaranteeing a high level of safety and privacy of sensitive data.
  - Robotics
    - from perception and decision-making to control.
    - Federated Learning is applied to improve multi-robot navigation under limited communication bandwidth scenarios, which is a current challenge in real-world learning-based robotic tasks.
    - Federated Learning is used to learn Vision-based navigation, helping better sim-to-real transfer.

Federated Learning - Challenges

- Dealing with non-IID (non-Independently and Identically Distributed) data
- Communication overhead
- Synchronization issues
- Heterogeneity between the different local datasets: each node may have some bias with respect to the general population, and the size of the datasets may vary significantly
- Temporal heterogeneity: each local dataset's distribution may vary with time
- Interoperability of each node's dataset is a prerequisite
- Each node's dataset may require regular curations
- Hiding training data might allow attackers to inject backdoors into the global model
- Lack of access to global training data makes it harder to identify unwanted biases entering the training e.g. age, gender, sexual orientation
- Partial or total loss of model updates due to node failures affecting the global model
- Lack of annotations or labels on the client side
- Heterogeneity between the processing platform

Ref: Federated Learning

a. https://federated.withgoogle.com/#:~:text=The%20federated%20learning%20approach%20for,and%20one%20driven%20by%20AI.
b. https://en.wikipedia.org/wiki/Federated_learning
c. https://arxiv.org/pdf/1902.04885.pdf
d. https://arxiv.org/pdf/1903.10635.pdf
e. Code: https://proandroiddev.com/federated-learning-e79e054c33ef

Meta Learning

Meta-Learning

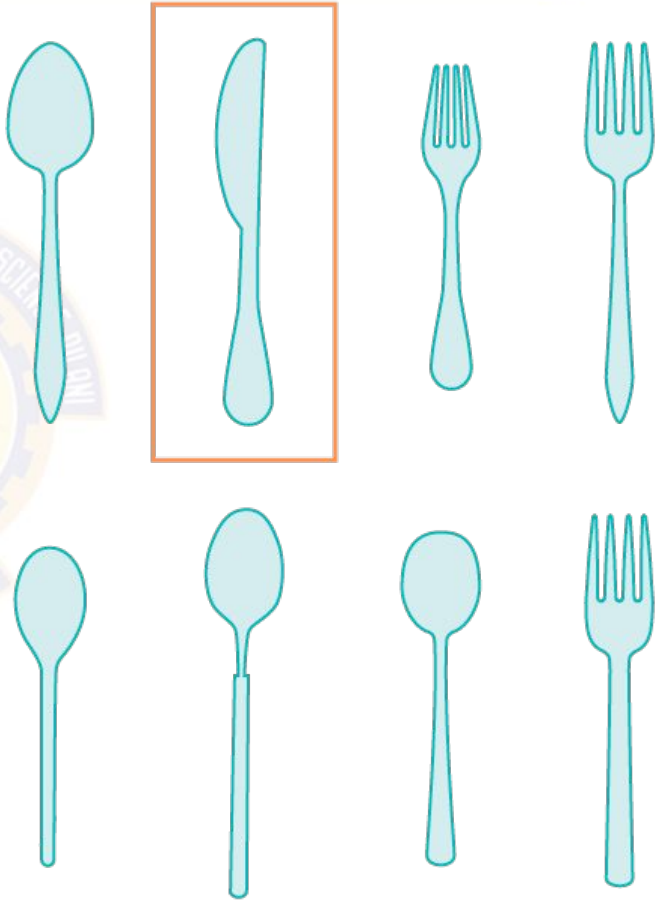
- **Learning to learn**
- Focus on developing algorithms and techniques that enable models to learn and adapt quickly and efficiently from a limited amount of data or across different tasks.
- The core idea behind meta-learning is to **improve the learning process** itself by leveraging prior learning experiences to tackle new, unseen tasks effectively.
- A model not only learns from a handful of examples, but also learns to classify novel classes during model inference.
- Learning with Limited Labeled Data

Meta-Learning

	Shallow Learning	Deep Learning	Meta-Learning
Classifier	**Learned**	**Learned**	**Learned**
Feature	Hand-crafted	**Learned**	**Learned**
Learner	Hand-crafted	Hand-crafted	**Learned**

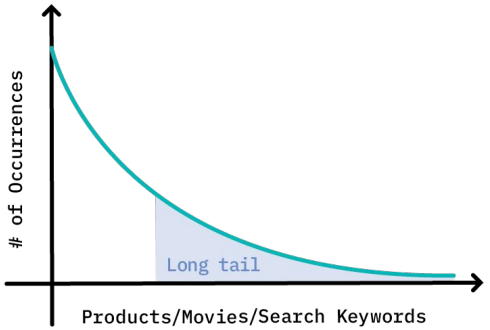
Meta-Learning

- Humans learn quickly.
- Humans adapt to new environments quickly.
- Humans leverage knowledge acquired from prior experience to solve novel tasks.
- Meta-learning leverages previous knowledge acquired from data to solve novel tasks quickly and more efficiently.

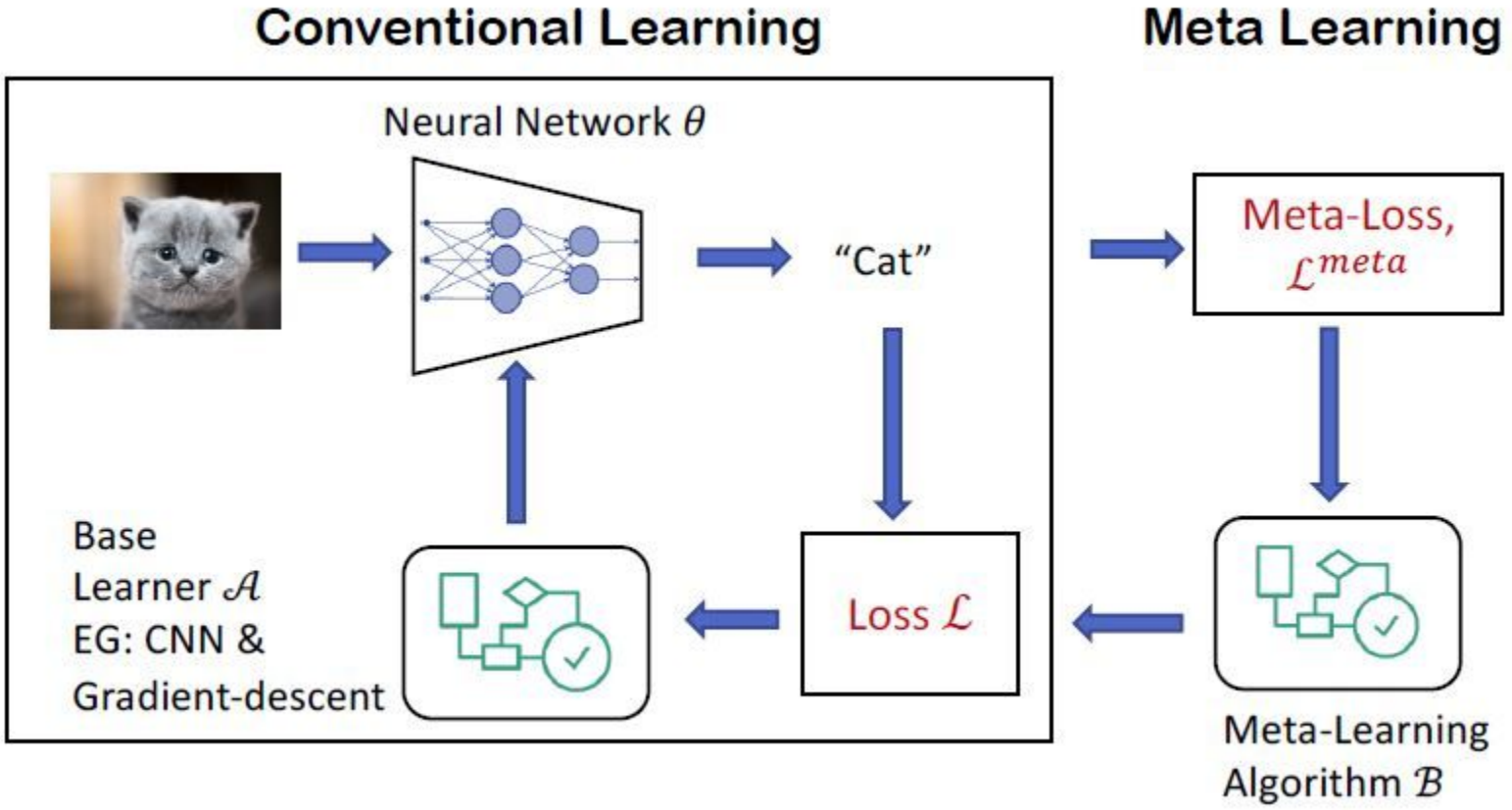


The problem of Generalization

- Supervised learning through deep learning methods requires massive amounts of labeled training data.
- The datasets are expensive to create, especially when one needs to involve a domain expert.
- Pre-training or transfer learning become less effective for domain-specific problems, which still require large amounts of task-specific labeled data to achieve good performance.
- Certain real world problems have long-tailed and imbalanced data distributions.
    - search engines: a few keywords are commonly searched for, whereas a vast majority of keywords are rarely searched for.



Meta-Learning



Meta-Learning

- **Learner** trains a model such as a neural network to solve a task such as visual object recognition.
- **Meta-learner** then wraps the learning process and trains the learner to better solve learning tasks.
- The **learner** typically minimizes a loss function that measures the difference between the true label of an input and the label predicted by a neural network model.
- The **meta-learner** minimizes a meta-loss function such as the generalization error of the trained neural network on novel data.

Why Meta-Learning

1. Ability to learn from a handful of examples
2. Learning or adapting to novel tasks quickly
3. Capability to build more generalizable systems.
4. Applications that require data-efficient approaches
    a. robots are tasked with learning new skills in the real world, and are often faced with new environments.

Key Aspects of Meta-Learning

1. Learning to Adapt:
   a. In meta-learning, the goal is to train a model to be highly adaptable, so it can rapidly adapt to new tasks with minimal additional training.
   b. In traditional machine learning, models are trained for specific tasks and require a substantial amount of labeled data.
2. Few-shot or Zero-shot Learning:
   a. Meta-learning often deals with scenarios where a model needs to learn from very few examples (few-shot learning) or even no examples (zero-shot learning) for a new task.
   b. The model leverages knowledge gained from previous tasks to generalize effectively.
3. Task Similarity:
   a. Meta-learning assumes that there is some degree of similarity or structure among the tasks encountered.
   b. Models are designed to capture this task similarity and transfer knowledge accordingly.

Meta-Learning - Three Approaches

1. Model-based
   - using (cyclic) networks with external or internal memory
   - models updates its parameters rapidly with a few training steps, which can be achieved by its internal architecture or controlled by another meta-learner model.
   - Memory-Augmented Neural Networks - encode new information quickly
   - Meta Networks - learns a meta-level knowledge across tasks
2. Metrics-based
   - learning effective distance metrics
   - Convolutional Siamese Neural Network - twin networks whose output is jointly trained to learn the relationship between input data sample pairs.
   - Matching Networks - learn a network that maps a small labelled support set and an unlabelled example to its label
3. Optimization-based
   - explicitly optimizing model parameters for fast learning
   - LSTM Meta-Learner
   - Model-Agnostic Meta-Learning (MAML) - general optimization algorithm
   - Reptile - meta-learning optimization algorithm

Meta-Data

Two levels of data:

- Meta-Training Data:
  - This data consists of multiple tasks with associated training data.
  - The model learns from these tasks.
- Meta-Test Data:
  - After meta-training, the model is evaluated on new tasks for which it has not seen the training data.

Steps in Meta-Learning

1. Meta-Training:
   - Task Generation:
     - Create a dataset of tasks, each defined by a task description and a few examples.
   - Task Sampling:
     - Randomly select a task from the dataset of tasks.
   - Training:
     - Train a base model (often called the "learner" or "base learner") on the examples provided by the selected task.
2. Meta-Testing:
   - Task Adaptation:
     - Select a new task from the dataset of tasks for which the model was not directly trained.
   - Fine-Tuning:
     - Use the few examples provided by the new task to fine-tune the base model.
   - Inference:
     - Make predictions or perform the task using the adapted model.

Steps in Meta-Learning

3. Evaluation:
   - Measure the performance of the adapted model on the new task.
   - Repeat the meta-testing process for multiple tasks to assess the model's ability to adapt across a range of tasks.
4. Meta-Learning Objective:
   - Define an objective or loss function that encourages the model to learn useful representations or adaptation strategies during the meta-training process.
   - This objective typically includes both the base model's training loss on each task and a regularization term that encourages the model to adapt well to new tasks during meta-testing.

Steps in Meta-Learning

5. Optimization:
   - Use optimization algorithms, such as gradient descent or reinforcement learning techniques, to update the model's parameters during meta-training.
   - During meta-testing, employ optimization algorithms like gradient descent to fine-tune the base model on new tasks.
6. Hyperparameter Tuning:
   - Tune hyperparameters of the meta-learning algorithm to achieve better adaptation performance.
   - These hyperparameters might include learning rates, regularization strengths, or architectural choices.

Steps in Meta-Learning

7. Regularization Techniques:
   - Apply regularization techniques to prevent overfitting during meta-training and encourage model generalization during meta-testing.
   - Common regularization methods include weight decay, dropout, and episodic training.
8. Model Architecture:
   - Choose an appropriate base model architecture that can be effectively adapted to various tasks.
   - Architectures can range from simple models like linear regressors to complex neural networks.

Steps in Meta-Learning

9. Dataset Preparation:
    - Curate or generate datasets of tasks and examples that are representative of the problem domain.
    - Ensure that tasks exhibit sufficient diversity and complexity to challenge the meta-learning model.
10. Iterative Process:
    - Meta-learning is often an iterative process where the model is trained on a sequence of tasks, gradually improving its adaptation capabilities.

Transfer Learning vs. Meta-Learning

- Both transfer learning and meta-learning involve leveraging prior knowledge
- Transfer learning fine-tunes a pre-trained model on a related task.
- Meta-learning trains a model to be able to adapt to various tasks from scratch, often with a minimal number of examples.

Meta Learning Libraries

- Torch-meta
  - https://github.com/tristandeleu/pytorch-meta
- learn2learn
  - https://github.com/learnables/learn2learn
- Meta-datasets
  - https://github.com/google-research/meta-dataset

Applications of Meta-Learning

1. Computer Vision:
   a. Few-Shot Image Classification: Meta-learning can be used to train models that can recognize new object categories with just a few examples.
   b. Object Detection and Segmentation: It can help in adapting object detection or segmentation models to new classes or domains.
2. Natural Language Processing (NLP):
   a. Transfer Learning for Text: Meta-learning can be applied to tasks such as sentiment analysis, language modeling, and named entity recognition to adapt models to specific domains or languages.
   b. Machine Translation: Meta-learning can facilitate adapting translation models to new language pairs.
3. Robotics:
   a. Robotic Control: Meta-learning can enable robots to adapt their control policies quickly to handle new environments or tasks.
   b. Object Manipulation: Robots can use meta-learned skills to manipulate objects with different shapes and sizes.
   c. Autonomous Vehicles:
   d. Adaptive Driving: Meta-learning can help autonomous vehicles adapt to changing road conditions or new traffic scenarios with limited real-world data.

Applications of Meta-Learning

6. Healthcare:
   a. Disease Diagnosis: Meta-learning can assist in building diagnostic models that adapt to the characteristics of specific patient populations.
   b. Drug Discovery: It can facilitate the adaptation of drug discovery models to new chemical compounds or biological targets.
7. Recommendation Systems:
   a. Personalization: Meta-learning can be used to personalize recommendation algorithms, adapting them to individual user preferences.
8. Finance:
   a. Financial Forecasting: Meta-learning models can adapt to changing market conditions for improved financial forecasting.
   b. Fraud Detection: Meta-learned fraud detection models can quickly adapt to new types of fraudulent activities.
9. Education:
   a. Adaptive Learning: In educational technology, meta-learning can adapt tutoring systems to individual student learning styles and needs.
10. Game Playing:
    a. Reinforcement Learning: Meta-learned reinforcement learning agents can adapt to new game environments or strategies efficiently.

Applications of Meta-Learning

11. Drug Discovery:
    a. Molecule Generation: Meta-learning can be applied to generate novel molecules with desired properties by adapting generative models to specific chemical spaces.
12. Meta Reinforcement Learning:
    a. Robotics Control: In robotics, meta-RL agents can adapt to various tasks or environments by quickly learning new policies.
13. Anomaly Detection:
    a. Network Security: Meta-learning models can adapt to changing cybersecurity threats and identify anomalies in network traffic.
14. Language Generation:
    a. Text Generation: Meta-learning can help text generation models adapt to specific writing styles or content domains.
15. Health Monitoring:
    a. Patient-Specific Monitoring: In healthcare, meta-learning can adapt monitoring models to individual patients' health data.
16. Human-Robot Collaboration:
    a. Cobotics: Robots can adapt their behavior to collaborate more effectively with humans in various contexts.

Ref: Meta Learning

a. https://arxiv.org/pdf/2004.05439.pdf
b. https://meta-learning.fastforwardlabs.com/
c. https://en.wikipedia.org/wiki/Meta-learning_(computer_science)
d. https://research.samsung.com/blog/Meta-Learning-in-Neural-Networks

Online / Incremental Learning

Incremental Learning

Ref: Online Learning

a. https://arxiv.org/pdf/1802.07569.pdf
b. https://arxiv.org/pdf/1904.07734v1.pdf

Thank you
All the best for the future :)