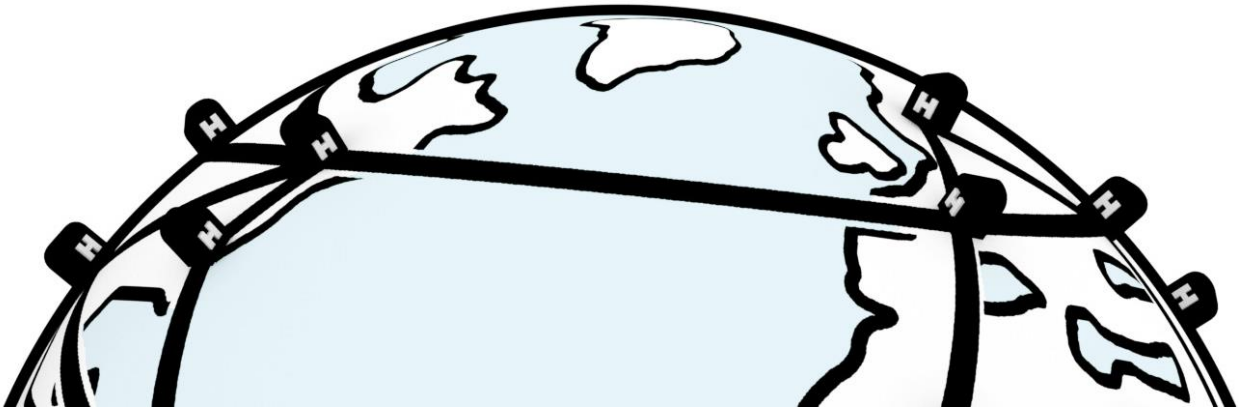
# Split Learning in Health Care

Multi-center Deep Learning without sharing patient data



Master's Thesis to achieve the degree of Master of Science in the Medical Imaging & Interventions specialization of the Technical Medicine program at the University of Twente.


Author: Maarten G. Poirot, BSc.[1,2,*]

Graduation committee:

| | |
|---|---|
| Chair and technical supervisor: | Ferdinand van der Heijden, PhD[1,3] |
| Medical supervisor: | Rajiv Gupta, MD, PhD[2,4] |
| Process supervisor: | Elyse M. Walter, MSc.[1] |
| External technical supervisor: | Praneeth Vepakomma, MSc.[5] |
| External member: | Anique T.M. Bellos-Grob, PhD[1] |


Colloquium: February 10th, 2020 3:00 pm CET


[1] Technical Medical Center, University of Twente, Enschede, The Netherlands
[2] Department of Radiology, Massachusetts General Hospital, Boston, Massachusetts, USA
[3] Department Robotics and Mechatronics, University of Twente, Enschede, The Netherlands
[4] Harvard Medical School, Boston, Massachusetts, USA
[5] Media Lab, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA
[*] correspondence: maartenpoirot@gmail.com

The work in this thesis was conducted at the department of Radiology of the Massachusetts General Hospital, Boston, Massachusetts, USA in collaboration with the Camera Culture Lab, Media Lab, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, between April 2019 and February 2020.

.

# Abstract

Split Learning obviates data sharing in multi-center Deep Learning. This work demonstrates feasibility of Split Learning for medical applications and introduces novel adaptations.

Deep Learning based models have shown to aid diagnosis, treatment and clinical workflow. But they rely on large quantities of diverse data and ever-increasing logistical resources and expertise often not available in single health care institutions. Furthermore, collaboration is hindered by regulatory, logistical and ethical concerns. This renders collaboration of multiple institutions infeasible for most studies, and leaves medical predictive models underperforming or undeveloped.

A solution, in the form of secure multi-party computing, has only been introduced recently. It does not require centralization of data but has yet to see adaptation in health care. Split Learning is a novel method in which a neural network is split into sequential elements that can be either private and distributed, or centralized, while retaining their functionality. This allows for configurations that do not require data nor label sharing. Feasibility and preference of this method over alternatives, but also opportunities for innovation are highly domain dependent and has not been researched in literature.

For the first time in literature, this work demonstrates Split Learning for clinical applications. We demonstrate feasibility and scalability of Split Learning for medical deep learning by comparing four major performance characteristics using four representative use cases. Additionally, we demonstrate several opportunities for innovation employing Split Learnings modularity for handling heterogeneous data, improving security and handling data streams from multiple institutions.

We conclude that the Split Learning paradigm meets all requirements for clinical feasibility while providing improved performance and reduced institution-side computational requirements compared to alternative methods. further opportunities for research into beneficial adaptations for clinical applications.

Secondly, our proposed method for handling heterogeneous data using Local Adapters shows promising initial results but requires further investigation. We analyzed the performance cost of increasing privacy and present Split Learning as a tool to redefine the concept of medical data.

# Samenvatting

Split Learning maakt het delen van patiëntgegevens in multicenter Deep Learning overbodig. Dit werk demonstreert de haalbaarheid van Split Learning voor medische toepassingen en introduceert nieuwe mogelijkheden.

Deep Learning modellen zijn nu al van grote toegevoegde waarde bij diagnose en behandeling. Voor de ontwikkeling van deze modellen is grote hoeveelheden data, logistiek en expertise nodig die vaak niet beschikbaar is voor individuele zorginstellingen. Bovendien wordt samenwerking gehinderd door regelgeving en logistieke en ethische problemen. Dit maakt multicenter onderzoek meestal onhaalbaar waardoor deze modellen niet optimaal, of überhaupt niet ontwikkeld kunnen worden.

Voor Deep Learning toepassingen enkele methoden recent geïntroduceerd die geen centralisatie van data vereisen. Deze zijn echter nog niet in de praktijk gebracht in het medisch domein. Split Learning is één van deze nieuwe methoden, hierbij word een neuraal netwerk wordt opgesplitst in opeenvolgende elementen die privé kunnen en gedistribueerd of gecentraliseerd kunnen zijn. Dit maakt het mogelijk modellen te trainen zonder dat data of labels gedeeld hoeven te worden. Toepasbaarheid en hoe het zich verhoudt ten opzichte van alternatieve methoden is sterk afhankelijk van het domein van toepassing en is tot op heden nog voor geen domein onderzocht.

Daarom presenteert dit werk voor het eerst een toepassing van Split Learning voor medische toepassingen. We demonstreren de haalbaarheid en schaalbaarheid van Split Learning door vier belangrijke kenmerken te vergelijken met behulp van vier representatieve toepassingen. Daarnaast demonstreren we verschillende mogelijkheden voor innovatie voor het omgaan met heterogene data, het verbeteren van de beveiliging en het verwerken van gegevensstromen vanuit meerdere instellingen.

We concluderen dat Split Learning voldoet aan alle vereisten voor klinische haalbaarheid, terwijl het verbeterde prestaties en verminderde computationele vereisten voor de instelling biedt in vergelijking met alternatieve methoden. Ten tweede toont onze voorgestelde methode voor het verwerken van heterogene gegevens met behulp van lokale adapters veelbelovende resultaten dat verder onderzoek vereist. We analyseerden de effecten van het verhogen van de privacy en presenteerden Split Learning als een hulpmiddel om het concept van medische gegevens opnieuw te definiëren.

# Acknowledgements

First and foremost, I would like to thank Rajiv Gupta for his guidance throughout this and previous projects. Without his limitless creativity and understanding of technology in health care this project would not have been possible. I would also like to thank all participants of the Split Learning group, -especially Praneeth Vepakomma, Ken Chang, Abhishek Singh, Vivek Sharma and Ramesh Raskar- at the Camera Culture group of the Massachusetts Institute of Technology Media Lab for the catalysis of countless possibilities, applications and opportunities of this paradigm. I would also like to thank my fellow students at New Chardon and Chandler street for making my time in Boston so vibrant. Lastly, I would like to thank my family, for standing by and supporting me throughout the years.

# Contents

# List of figures

# List of tables

# List of Acronyms

| | |
|---|---|
| AI | artificial intelligence |
| ACR | American Society of Radiology |
| AUROC | area under receiver operator characteristic |
| AutoML | automated machine learning |
| BraTS | brain tumor segmentation |
| CheXpert | Large Chest X-ray expert data set |
| CI | confidence interval |
| CIIL | cyclic institutional incremental learning |
| CNN | convolutional neural network |
| CT | computed tomography |
| CWT | cyclic weight transfer |
| DL | deep learning |
| DML | distributed machine learning |
| DRC | diabetic retinopathy challenge data set |
| EHR | electronic health records |
| FCL | fully connected layer |
| FL | Federated Learning |
| FLAIR | fluid attenuated inversion recovery |
| GDPR | General Data Protection Regulation |
| GE | General Electronics |
| GPU | graphical processing unit |
| HIPAA | Health Insurance Portability and Accountability Act |
| IID | independently and identically distributed |
| IIL | institutional incremental learning |
| KL | Kullback-Leibler |
| ML | machine learning |
| NLP | natural language processing |
| PET | positron emission tomography |
| MRI | magnetic resonance imaging |
| ResNet | residual network |
| SL | Split Learning |
| SMPC | secure multi-party computing |

# List of Symbols

| | |
|---|---|
| $K$ | number of participating institutions |
| $N$ | total number of model parameters, a sum of $N_{front}$, $N_{center}$ and $N_{back}$. |
| $p$ | total data sets size $p$ in bytes |
| $q$ | total size of layers sent over $q$, a sum of $q_{front}$ and $q_{back}$ in bytes |
| $\eta$ | fraction of total number of model parameters residing locally |
| $\Omega$ | communicational overhead |
| $\tau$ | computation time per batch |
| $v$ | bandwidth speed |
| $X, Y$ | raw data and target labels |
| $F^h_{front}$ | neural network named $front$ located at $h$ |
| $L_n$ | $n^{\text{th}}$ neural network layer |
| $X_n$ | features produced by $n^{\text{th}}$ neural network layer |
| $\nabla$ | gradient |
| $\hat{Y}$ | predicted labels |

# 1 Introduction

## 1.1 Motivation

Deep learning based predictive models have already shown to be of great benefit in automation and standardization of clinical decision making in diagnostics and therapy.[1] Common deep learning based medical tasks include image classification[2], speech recognition[3] and natural language processing[4] based on complex, high dimensional and sensitive data such as electronic health records (EHR), diagnostic imaging, biosensors, omics and text.

These networks rely on vast amounts of diverse, structured training data in order to converge, reach optimal inference performance, generalize and be robust.[5,6] However, medical sample sizes in single institutions tend to be small, especially in less prevalent diseases and diseases with less standardization of care.[7] In addition, it has been observed that the number of model parameters can drastically improve performance with the largest models reaching in the billions of parameters.[8–11] This increase in depth and complexity of deep learning models requires evermore expertise and computing power that is not available to most institutions.

Collaboration among institutions holds the key to resolve these problems by increasing the amount of available data and its diversity, and centralizing training effort. But the required data centralization forms a barrier through regulatory, ethical and logistical constraints.[12–15] For one, regulations to protect patient privacy such as HIPAA and GDPR usually restrict even anonymized patient data to leave the premise as anonymization alone is inadequate to prevent re-identification.[16] In rare cases, patients could for example be identified based on disease status and scanning region[17,18]. Secondly, policy to protect institution property, or even unwillingness to share this valuable commodity can obstruct centralized pooling of data as it reduces level of control.[19] Thirdly, data often lacks the appropriate consent. Lastly, centralized solutions impose logistical challenges that require funding and expertise such as additional file-server storage and bandwidth requirements. This often renders multi-center studies infeasible and results in only 2.9% of published machine learning studies to include data from multiple institutions[20], leaving value in data locked off that could have been employed to improve clinical decision making.

Methods for multi-center machine learning without data sharing have shown to solve this problem in training models in data centers and on mobile devices[21] carrying sensitive information. These distributed machine learning (DML) methods allow for training of predictive models without

centralization of data. This lowers the entry barrier for data providers to collaborate. Thus, mitigates previously mentioned obstructions, while retaining the benefits of a larger, more diverse data pool.

DML as a concept has seen applications in, for example, mobile devices.[21,22] However, its application in health care is currently limited to proof-of-concept studies of a few methods. In this work we consider one of the most recently introduced DML method for health care, namely Split Learning[23], and compare it to the most popular alternative, Federated Learning[24]. The concept of Split Learning is the splitting of a neural network into several sequential elements. This allows part of the network to remain distributed, and others to be centralized. The network can then be trained by sending intermediate network activations from distributed sources and a centralized server. In comparison, Federated Learning aggregates locally trained models on a centralized server. When compared, Split Learning provides interesting traits such as its modularity, identical functionality to centralized data, low institution-side computational cost, and a fundamentally different communication cost equation.

## 1.2 Contribution

For the first time in literature, this work demonstrates Split Learning for medical applications.

In the first part about clinical feasibility of Split Learning, we propose, discuss, and implement novel adaptations to conventional neural networks that enable Split Learning. We then investigate performance aspects including inference performance, convergence rate, computational efficiency, and communicational requirements, and translate this to clinical feasibility. We do this using several representative medical imaging tasks for a variety of scenarios such as range of participating institutions or variable dataset sizes.

In the second part, we propose, and test several adaptations to Split Learning based on observations made in the first part, to display future opportunities of the paradigm. The first proposal aims to account for inter institutional data heterogeneity using domain adaptation. The second improved security using alternative weight sharing strategies. And the last aims to enable making single predictions from multiple data sources, also known as vertically partitioned data.

## 1.3 Outline

In this work, Split Learning is implemented, developed, applied and evaluated for different medical tasks. Evaluation takes in account several performance factors, which are qualitatively compared to alternatives, and translated to their meaning in health care implementations. In addition, we propose and apply modifications to the default concept of Split Learning aimed combating data heterogeneity, security and logistical challenges.

In chapter 2 the background of machine learning in health is discussed. It poses both the motivation as basis of reasoning for the rest of the work. First, the added value and a range of

implementations of machine learning in health care are outlined. Secondly, its inner workings will be discussed, aimed at providing the reader with the background used in later chapters. Lastly, challenges of machine learning implementation in health care and their link to secure multi-party computing will be discussed.

Chapter 3 discusses secure multi-party computing and several strategies of enabling multiple institutions to collaborate without sharing their sensitive data. In the second part, distributed machine learning is introduced in combination with descriptions of several existing methods. We will introduce Split Learning in depth and highlight features that are of interest for health care implementation specifically and the research that they would require.

Chapter 4 investigates feasibility of Split Learning in health care. It proposes the methods of analysis for the various performance characteristics of interest on vanilla Split Learning. It describes the testbed that is created to run Split Learning experiments on. It describes in detail the tasks, data sets, and neural network topologies that were implemented to analyze inference performance, convergence rate, computational efficiency, and communication overhead for a range of scenarios. And finally, it provides a verdict about the feasibility of Split Learning, preferent cases for Split Learning implementation in the context of alternative methods and training on centralized data.

Beyond opportunities described in the previous chapter, Chapter 5 proposes, investigates, and discusses modifications of Split Learning. These modifications include varying communication strategies among institutions to improve privacy and a domain adaptation strategy that employs the modularity of Split Learning to account for inter institutional heterogeneity.

A concluding chapter on both topics is provided at the very end of this work.

# 2 Machine Learning in Health Care

Machine learning is a subset of artificial intelligence (AI) that learns to perform tasks from training data as oppose to explicit programming. In 2019, machine learning was the top area of innovation in health care[25]. At the same time, health care demonstrates the greatest application challenges for this method.[26,27] An understanding of both the field and the method is required to efficiently address inhibiting factors for machine learning to progress in health care. This chapter will provide a basis for understanding how the opportunities for innovation in health care match the capabilities that machine learning can provide. We will first describe the need in health care and how machine learning has already shown to be capable of addressing these needs. To solidify our understanding of the capabilities of machine learning, this chapter will provide the scientific fundamentals to understand its strengths. Finally, we will be able to examine the key challenges that would have to be addressed to ensure long-term success.

## 2.1 The promise of machine learning in health care

Machine learning is hoped to aid health care in meeting the increasing demand and complexity of care and provide new insights to realize improvement of care. This has been enabled by developments in computing power and the huge volumes of data that are being generated. The combination of these factors has put machine learning-based solutions at the forefront of health care innovation.

### 2.1.1 Rising demand for health care

Most countries are seeing their life expectancy rise, and their population age.[28] These older citizens require more health care and expect a different, more personalized health care treatment.[29] In conjunction, more ailments, diseases and disorders are becoming treatable, which increases the population suffering from chronic diseases.[30] Future patients will present more coexisting illnesses and medications increasing the call for personalized medicine and making medical care more complex.[31] All these factors contribute to a steadily increasing demand for health care[32] and increasing medical complexity. This rising demand for health care has already pushed the workload per clinical visit beyond the time allotted per visit and can only be expected to worsen.[33,34]

### 2.1.2 Change of medical professions

Technological progress has changed the type of work and workload clinicians. Whereas radiologists were once confined to two-dimensional projection images such as chest radiographs, they now

possess a range of complex and high-dimensional imaging modalities. Although cross-sectional imaging such as computed tomography (CT) and magnetic resonance imaging (MRI) have enabled more accurate diagnoses[35,36], they come at a price of an increased amount of data that has to be reviewed. Radiologists are reading an increasing number of cases with more images per case. This abundance of data has changed how radiologists interpret images. Work has changed from pattern recognition, with clinical context, to searching for needles in haystacks; from inference to detection. These workloads are so demanding that fatigue may impact diagnostic accuracy[37–40] and physician shortages further exacerbate the problem, especially for radiologists in medically underserved areas.[41] In the case of the diabetic retinopathy epidemic[42] in India, integration of these systems has already allowed remote clinics to provide screening of fundus photographs with quality comparable to a board of medical experts. This is especially valuable in these remote areas as ophthalmological expertise can be lacking.[43,44]

The amount of information continues to increase in imaging, both extractable by the human eye and extractable only by software.[45] The abundance and complexity have empowered, but also challenged, medical experts in clinical decision-making. This has paved the way for the role of computers, which extract fine information invisible to the human eye and process those data quickly and accurately. It is not expected that AI will replace physicians in this process, but it could empower physicians in the bulk of routine work and shift focus back to the human side of medicine.[46,47]

### 2.1.3 Systematic improvement

The appeal of cognitive automation extends beyond reduction of labor into reducing the presence of human error. Arriving at a medical diagnosis is a highly complex process that is extremely error prone.[48] Medical imaging is a major contributor to the overall diagnostic process, but also a major potential source of diagnostic error. Most medical diagnoses are not missed because of the technical or physical limitations of the imaging modality, but because of image interpretation errors by radiologists.[49] Other inadvertent human errors like prescription overdoses[50,51], result from the intrinsic lack of standardization in human decision making. Whereas the overall prevalence of radiologists' errors in practice does not appear to have changed since it was first estimated in the 1960[49], computer based method provide handles for systematic improvement.

### 2.1.4 Novel opportunities

It is this systematic improvement that forms the final merit. Guided by relevant clinical questions, powerful AI techniques can unlock clinically relevant information hidden in the massive amount of data, which in turn can assist clinical decision making.[52–54] Machine learning differs fundamentally from manual methods in both discovery of new features as detection of found features Manual feature discovery by doctors consists can be empirical and make use of domain knowledge and reasoning, whereas machine learning based feature detection can detect many features systematically. The task of reading these features is also fundamentally different. Consider the challenge of

reading electrocardiograms. Conventional feature discovery consists of understanding of the underlying physiology of the heart and empirical evidence. To contrast, discovery and diagnosis using machine learning algorithms consists of systematically analyzing every heartbeat. There are early signs that such analyses can identify subtle microscopic variations linked to sudden cardiac death.[55] Another example is voice, as it has shown to contain features that distinguish health from depressed people[56] but that have been hard to quantify. Machine learning could provide objective criteria of psychomotor retardation or more general vocal acoustic features as biomarker for mental health disorders.[57,58] Even more profound are aims to identify or redefine subgroups in heterogeneous diseases using features unaccounted for by contemporary diagnostic methods.[59,60] These examples are by no means extensive, but they do support the hope that machine learning will enable better disease surveillance, facilitate early detection, allow for improved diagnoses, uncover novel treatments, and allow for more personalized medicine.

## 2.2 Scientific fundamentals of machine learning

### 2.2.1 The field and its subsets

No single definition exists for the overarching term 'AI'. The term is most commonly used for algorithms that mimic functions associated with human cognitive functions like 'learning', 'recognizing' and 'problem solving'. Because this definition also includes tasks that might be considered of trivial complexity —like finding the shortest route through a maze— it is noted that the definition of AI shifts every time we figure a piece out and we say "that's not thinking"[61]. Counterintuitively, the hardest problems for AI to solve are often times the easiest for humans. These are the problems that we solve intuitively but are hard to describe formally.[62] Machine learning, a subset of AI, targets these problem by learning relevant features from training data.

The field itself has existed for decades and is closely related to statistics and optimization. It has seen rapid advancements due to developments in computing power and the increase of digitalized information in the last few years.[62,63] The most popular machine learning technique in medicine is artificial neural networks.[64] These networks are computational analytical tools inspired by the biological nervous system. A neural network consists of artificial neurons that share characteristics with neurons in a biological brain. Like the synapses in a biological brain, neurons are densely interconnected and act as simple processing nodes: they transmit activations based on the processing of signals of their input. In artificial neural networks, the neurons are most commonly organized in sequential layers, that in combination with their connectivity define the network topology, also known as architecture. Increasing the number of layers increases the depth of the network. It has been shown that increased depth improves expressive power and potentially performance.[8–11,65] This has caused networks to become deeper and to consist of more neurons, creating the term "deep learning". For image processing, convolutional neural networks (CNN) are of most interest through their property to capture the spatial relations in an image through application of convolutional filters.

Convolutional neural networks have also been inspired by biological processes, specifically the connectivity pattern of the visual cortex.[62,66,67] The visual field is divided into smaller visual fields that partially overlap and are covered by individual neurons. In fully connected networks each neuron in one layer is connected to every neuron in the next. This increases the amount of connections exponentially and renders the network prone to overfitting. Convolutional neural networks are a regularized version of fully connected networks that reduce overfitting. In overfitting, predictions of the trained model correspond too closely to the training data, see figure 1. Regularization improves generalization and thus performance on newly presented data. Another benefit of convolutional neural networks is that they require relatively little preprocessing.

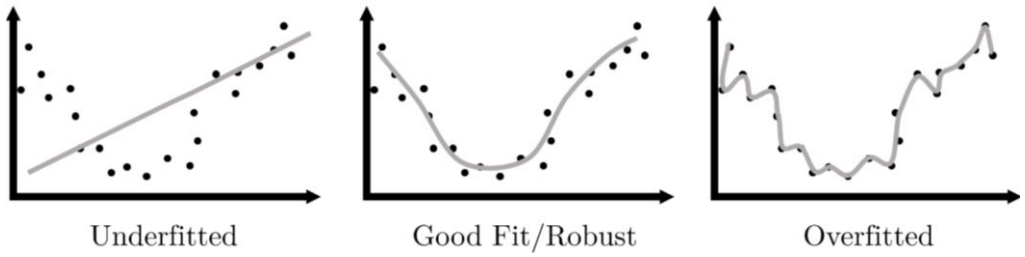



*Figure 1: Visual examples of model fitting. Overfitted models do not generalize well for new data.*

### 2.2.2 Neural Network Architecture

The core strength of artificial neural networks is that they do not rely on manually-crafted features.[68,69] Consider a supervised neural network which can be described as a function that takes data point $\boldsymbol{x}$ to make a prediction $\hat{\boldsymbol{y}}$ of label $\boldsymbol{y}$. This predictive function is also called the model. The model is comprised of millions of nonlinear neurons with adjustable weights and biases. The weights and biases of these neurons form the parameters of the network. These neurons are structured in sequential layers such that they only allow data to be processed in one direction. The configuration of these layers and their connections constitute the network *architecture*. Deep neural networks are characterized by a very large number of layers.

It is good to know that many other types of layers exist besides the fully connected and convolutional layers mentioned earlier, and most commonly visualized. Some layers are trainable, others add regularization or apply dimensionality reduction. The order and parameters of these many layers define the earlier mentioned architecture The architecture is part of the so called '*hyperparameters*'. The search space for these hyperparameters is immense and there is no way to guarantee an optimal configuration. Manual design of the architectures has therefore sometimes been called an art, rather than a science.[70] It is common practice, especially in medicine, to reuse an architecture that has been shown to work on a similar problem. Alternative options are methods for automation of the modeling process like automated machine learning (AutoML) and Bayesian hyperparameter optimization.[71]

### 2.2.3 Training a neural network

It is the weights and biases of these neurons, the parameters, that make up the full equation and thus the error between $\hat{y}$ and $y$, the accuracy of our prediction. It is therefore that the iterative optimization of these parameters is what we call *learning*. This optimization process is non-trivial, as the number of parameters in these networks often run into the millions. Brute force approaches like a grid searches are out of the question. The basis of optimization can be imagined as traversing the mountainous landscape in the space of weight values blindfolded. This might sound hard, but you should at least be able to reach a valley by following the descent in the landscape or put otherwise: reach a local minimum by following the negative gradient vector. Interestingly enough, poor local minima are rarely a problem for large networks and they nearly always reach solutions of very similar quality.[72]

One learning step consists of a forward pass, an error computation, and a backward pass. All these steps consist of matrix operations on the high dimensional matrices called *tensors*. In the forward pass a data tensor $X$ is passed through the layers, where the output of every layer is the input of the next. The bulk these layers use the output of the previous layer to derive representations that accumulate to contain higher semantic representations in deeper layers, this is visualized in Figure 2. The final representations are provided to the last layer, usually a classifier, that generates $\hat{y}$, completing the forward pass.[62] The error between $\hat{y}$ and $y$ is computed using a predefined *objective function* also called the *loss function*. The objective function is used to start the backpropagation process. It is important and task-specific as it determines how the difference between $\hat{y}$ and $y$ is defined, and thus what exactly should be learned. In the following backpropagation process the gradient of the objective function with respect to the state of the model is a practical application of the chain rule of derivatives.[27] It is repeatedly applied to propagate gradients through the network, back to front. Finally, the parameters of the network are updated and one training step has been completed.

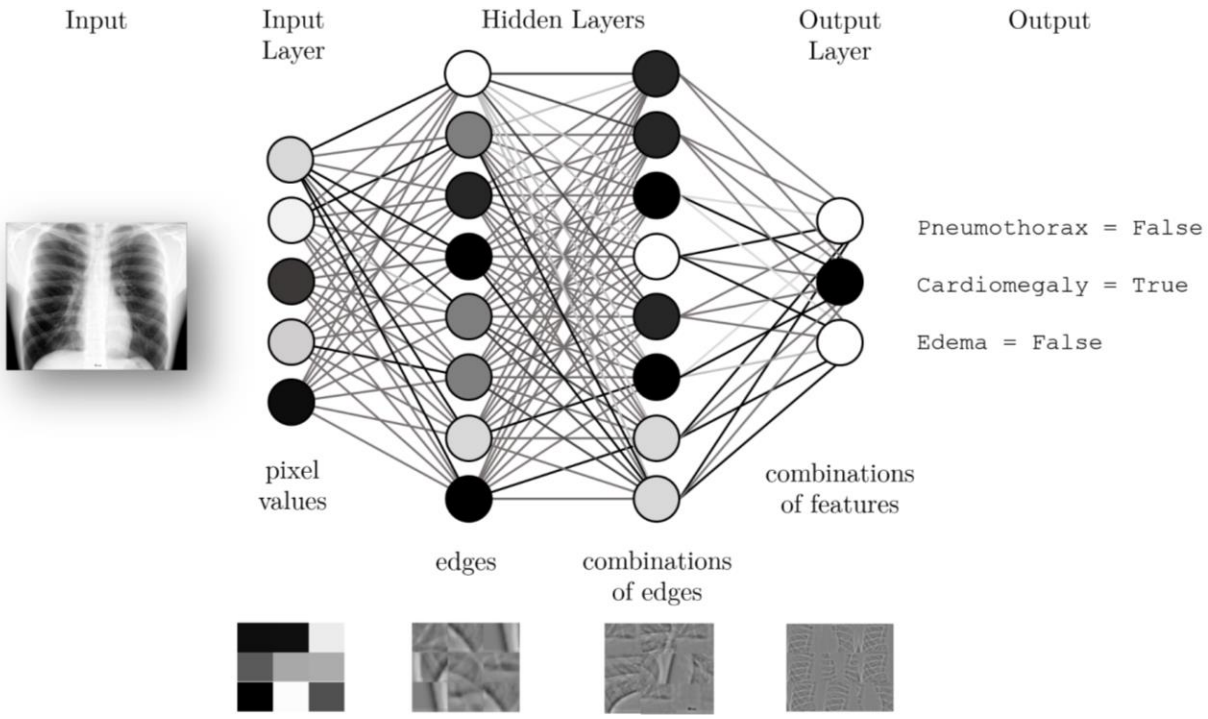



*Figure 2: Simplified graphical representation of a deep neural network with two hidden layers. Circles represent neurons, vertically aligned in layers. Lines denote inter layer connectivity, with darker lines suggesting varying weights. Deeper layers capture higher semantic content with examples provided below the graph. Input data is represented left, forward propagation runs left to right. Objective function is computed right, and backpropagation runs right to left.*

### 2.2.4 Machine Learning tasks

Machine learning algorithms can be broken down into their approach of dealing with data. Most belong to one of three types of learning algorithms: supervised, unsupervised and reinforcement learning. Due to its popularity we will limit ourselves to supervised deep learning tasks, although we expect our results to generalize for the other tasks as well.

#### *2.2.4.1 Supervised Learning*

The most common form of machine learning is supervised learning, also called inductive learning. It learns a set of underlying patterns from instances that should generalize for new instances.[68] These instances are explicit input-output pairs which incurs collecting a large labeled data set. Data is used to learn features that hold predictive power. In linear regression, these could be as simple as specific independent variables, but, in image classification, these features can be semantically complex, reducing transparency in solving these complex tasks.[27,73] Careful curation of these data sets is often required as resulting performance is strongly dependent on the quality of the data provided.[62] In addition, larger, more diverse data sets contain a more complete representation of the probability distribution. This increases predictive performance. Tasks most frequently performed by supervised learning algorithms are classification and regression problems. Medical examples of these kinds of tasks are given in figure 3.

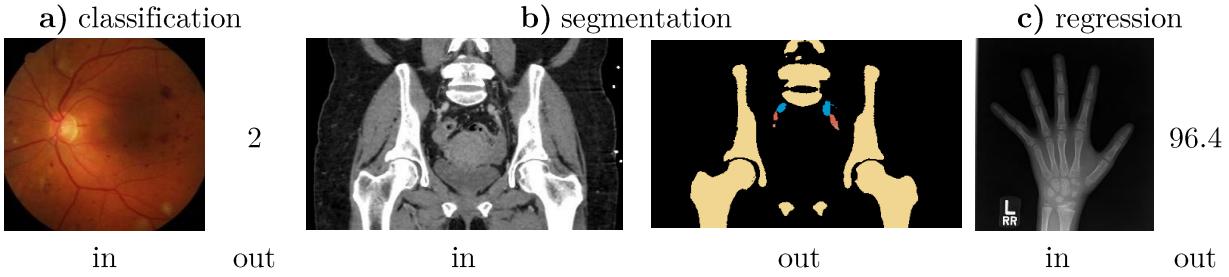



*Figure 3: Examples of typical supervised learning tasks. a) Staging of diabetic retinopathy from fundus photographs.[74] b) Segmentation of anatomy from abdominal computed tomography (CT) scans.[75] c) Determining skeletal age pediatric hand radiographs.[76]*

#### *2.2.4.2 Unsupervised Learning*

Unsupervised learning does not require these labels. It can perform operations like clustering, noise reduction or generating samples from a learned distribution. The benefit of this method is its total lack of dependence of information provided by labels. A drawback of this method is that the result generated can be hard to interpret, since it might not match any predefined semantic meaning. Medical examples of this approach are provided in Figure 4.

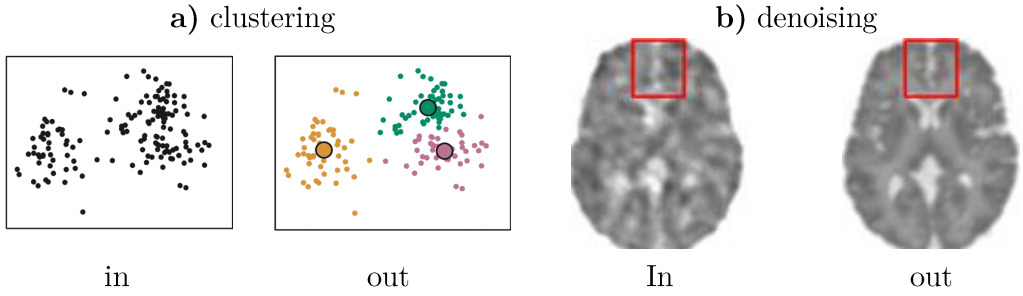



*Figure 4: Examples of typical unsupervised learning tasks. a) Identifying sub-populations of patients with cardiovascular disease who may benefit from different medication[77]. b) Positron emission tomography (PET) image denoising[78].*

#### *2.2.4.3 Reinforcement Learning*

Lastly, reinforcement learning is a method that gives rewards or punishments to train the algorithm in optimizing its strategy. This type of method is most often used to teach algorithms play games but has also shown to be useful for research in optimization of protein folding. However, it has seen relatively few clinical applications.

## 2.3 Main inhibiting factors

Machine learning is a technique that comes with its strengths and weaknesses. In conjunction, it's application in the medical domain comes with specific challenges. We discuss properties of both the technique and the domain to distill key challenges that should be faced to ensure rapid adaptation of machine learning in health care.

### 2.3.1 Medical data properties

Humans and animals are able to learn from very few examples. We do not yet know how this is possible.[62] In contrast, most current artificial neural networks rely on vast amounts of diverse, structured training data in order to converge, reach optimal inference performance, generalize and be robust.[5,6] However, most of the various types of data used in modern biomedical research like electronic health records, imaging, -omics, sensor data and text, are generally complex, heterogeneous, poorly annotated and unstructured. In addition, non-representative demographics have also demonstrated reduced performance on underrepresented groups.[79,80] When different institutions employ for example different scanners, scan protocols or even sequences, this attributes to inter-institution data heterogeneity. In addition, different labeling protocols can render label semantics incongruent. Both can deteriorate inference performance, regardless whether training is performed in distributed fashion. This effect is generally greater in uncontrolled environments like clinical practice than the more controlled research environments. Coordinated data-preprocessing and common labeling protocols should be employed minimize these effects.[81] However, medical sample sizes in single institutions tend to be small, especially in less prevalent diseases and diseases with less standardization of care.[82] The increase in access to high quality medical data is critical in the development and ultimate implementation of AI applications in health care.[1,5,7,25,83–87]

### 2.3.2 Medical data sharing

The promise of AI is tightly knit to the availability of relevant data. Even though there is an abundance of data in the health domain, the quality and accessibility of these resources remains a significant challenge.[84] Collaboration among institutions holds the key to resolve these problems by increasing the amount of available data and its diversity, and centralizing training effort.[88] But the required centralization of this sensitive and valuable information forms a barrier through regulatory, ethical and logistical constraints.[12–15] Initially, regulations such the Health Insurance Portability and Accountability Act of 1996 (HIPAA) in the United States or the General Data Protection Regulation (GDPR) in the European Union usually restrict even anonymized patient data to leave the premise to protect patient privacy. This is due to the observation that anonymization alone is inadequate to prevent re-identification.[16] In rare cases, patients could for example be identified based on disease status and scanning region[17,18]. Secondly, health care data is expensive to collect, especially in the cases of longitudinal studies and clinical trials. Therefore, policy to protect institution property, or even unwillingness to share this valuable commodity often obstruct centralization of data as it reduces level of control.[19] Thirdly, there is an ethical dilemma that data on the infinite reuse of the implicit consent given by patients. Lastly, centralized solutions impose logistical challenges that require funding and expertise such as additional file-server storage and bandwidth requirements. This often renders multi-center studies infeasible, leaving value in data locked off that could have been employed to improve clinical decision making.

### 2.3.3 Logistical requirements

Limiting factors can also have logistical origins. The level of abstraction achieved by these networks is in part by their vast amount of parameters, with the largest models reaching in the billions of parameters.[8–11] Most clinical institutions might not possess the compute to train or use these models.[8,9] In addition, these models might require a level of expertise to engineer and oversee that can be inhibitory for most hospitals to employ their data to train models.[23,90]

## 2.4 Conclusion

AI holds the potential to address critical health challenges and improve medical care. The effect neural networks can have in health care is highly reliant on the training data available. Even though there is an abundance of data in the health domain, the quality and accessibility of these resources remain a significant challenge due to regulatory and logistical constraints. The alleviation these constraints would open up vast amounts of information to implement these algorithms in more places and higher efficacy.

> *"It's not who has the best algorithm that wins.*
> *It's who has the most data."*
>
> *- Andrew Ng*

# 3 Privacy-Preserving Collaboration

The multi-center approach to conduct medical research holds many merits over conventional methods. However, centralization of data is a burden to set up. Alternative methods from the field of secure multi-party computation aim to allow collaboration without sharing sensitive data. Although most methods are not applicable to the computationally heavy demands of machine learning, few are. Some of these distributed machine learning methods and their properties related to security, inference performance, computational efficiency and communications cost will be highlighted. In conclusion, we will thoroughly introduce Split Learning, emphasize its interesting properties for medical applications and describe research that remains to be done.

## 3.1 Multi-center research

In health care, multi-center research is the cornerstone of clinical trials of treatment and diagnostic innovations for patient care. The larger sample size and more diverse population in multi-center clinical research can reduce the time needed to obtain the required number of study subjects, increase statistical power[91], and produce more results in terms of expertise and facilities.[92] To broaden the impact of machine learning research in health care, research should therefore be conducted multi-centrally.[93,94]

## 3.2 Secure Multi-Party Computation

Widespread application of neural networks in sensitive areas, such as finance and health, has created a need for both distributed and secure training and inference of neural networks called Multi-Party Computing (SMPC).[95–97]

The distributed component of these methods was first proven to be feasible in server grade distributed gradient optimization[98,99] to simply train large scale deep neural networks over multiple machines[100] or to efficiently utilize several GPUs on a single machine.[101] The security component differentiates itself from traditional cryptography tasks in not just assuring security and integrity of communication against adversaries outside the system, but also from each other. Under this paradigm the owner of the network does not require access to the actual raw data used to train the model.[102] Neural networks have shown to be very robust to addition of noise and have shown

to be able to recover imaging from partial imput[103] which makes it difficult to compute them securely.[104]

To fill this need, several methods have been developed of we examine a few. Initial methods used cryptographic primitives to provide privacy-preserving multi-party computation. In homomorphic encryption[105–107] a central computing node computes on encrypted data and also returns encrypted data. In oblivious transfer, a central computing node is oblivious to the data it receives, and the institution is oblivious to the computation performed.[108,109] However, these methods have shown to not scale well for deep learning as they are inherently inefficient.[109,110] Some viable distributed Deep Learning methods do exist, of which we will discuss Cyclic Weight Transfer, Federated Learning and Split Learning. Our main interest goes out to Split Learning due to several interesting properties we will introduce later on.

### 3.2.1 Cyclic Weight Transfer

Institutional Incremental Learning (IIL) is a very basic collaborative learning approach and forms the foundation of Cyclic Weight Transfer. In IIL, institutions train a model that is shared in succession. An institution receives the current state of the model, trains it on its own data, forwards it to the next and finally receives the model that was trained on by all institutions. Bandwidth usage is very low: only the model state is sent and received twice. The major drawback of this method is a drop in performance with an increasing number of institutions due to catastrophic forgetting.[111,112] Cyclic Weight Transfer[113] (CWT), also known by the name Cyclic Institutional Incremental Learning (CIIL)[81] alleviates this problem by fixing the amount of epochs each institution trains the model, before passing it on to the next. This can significantly increase communication cost, especially for larger models when cycling relatively often. In addition, a trade-off between communication efficiency and accepting a performance drop-off remains.

### 3.2.2 Federated Learning

In Federated Learning, a model is trained in a hub-and-spoke configuration. The hub is a centralized server that distributes a conventional neural network over its participating institutions. They train the model and after some time return their states to the central server. The central server performs an aggregation step of all these updates, after which the cycle is repeated. In Federated Learning, no raw data but only model sates are being shared. An advantage to CWT is possibility of parallelized training. The aggregation step performed does however incur a loss in performance, especially for cases with smaller institution-side data sets.

## 3.3 Split Learning

In its most general form, Spit Learning splits a conventional neural network into any number of components that can then reside locally and distributed, or centrally while retaining their function in the network. Although many configurations can be imagined, the scope of this work is limited to the simplest Split Learning configuration that does neither require data nor label sharing, called

the *Boomerang* configuration. This configuration was named *U-Shaped* in literature[23,110,114], but to avoid confusion with a type of commonly used deep neural network architecture called *U-Nets* we introduce the name *Boomerang Split learning* in consultation with the original authors. Boomerang Split Learning requires no input data sharing, no label sharing and allows a large part of the computational load to be centralized.

### 3.3.1 Definitions

To explain Boomerang Split Learning, we consider a setup that consists of multiple institutions that hold their proprietary data, and a single centralized server that holds no data. The paradigm of Split Learning revolves around splitting a conventional neural network $F$ consisting of layers $\{L_0, L_1, \dots L_N\}$ into sequential elements, analogous to a *chain* consisting of *links*. These links can then be *local*: located and accessed by its owning institution, or *central*: hosted by the central server and accessed as black box by all. In Boomerang Split Learning, the chain in this configuration consists of three links such that $F_{front}, F_{center}, F_{back} \leftarrow \{L_{0\to n}\}, \{L_{n+1\to m}\}, \{L_{m+1\to N}\}$. The links are named from a forward propagation point of view. The first is called *front* and is local. It receives raw input data during forward propagation and returns features of the $n^{\text{th}}$ layer. The second link is called *center* and is centrally hosted. It takes the features from the front, performs most of the computation and forwards another set of features from the $m^{\text{th}}$ layer to the final link called *back*. The back is again local and performs the final decoding computation on its input. This local stage is where gradients are computed from the decoded output and labels using a predefined objective function $G$. This configuration is visualized in Figure 5.

Typically, the largest part of trainable networks layers can be found in the center. This can reduce bandwidth used in sharing local states, as well as institution-side computational cost. This property should allow computation of more complex networks for institutions with less computational power, compared to alternatives like Federated Learning.

### 3.3.2 Training process

The training process consists of one institution feeding a sample of raw data $X$ to its front network $F_{front}$. The resulting activations $X_n$ are passed onward to the center, and consequently to the back as $X_m$. The back computes the output $\hat{Y}$. The output and corresponding labels $Y$ are used by the objective function $G(\hat{Y}, Y)$ to compute the loss and initiate back propagation. After back-propagating the gradients one data sample has been processed, all model states are updated, and a next sample-pair can be processed. Forward propagation in Split Learning involves sequential computation and transmission followed by computation of remaining layers, which is functionally identical to applying all layers at once. This also holds for the backpropagation process, due to the chain rule in differentiation. Thus, in contrast to alternative methods, Split Learning does not require any aggregation and is functionally identical to training in centralized fashion[23], and is expected to yield more similar performance results.

By default, training is switched to the next institution when the entire batch of a single institution has been processed, we call this mode *alternating-epochs.* This switch includes sending the state of the local links from the last trained institution to the next institution to update its states. The algorithm for this process is provided in appendix 8.2. This method could theoretically suffer from catastrophic forgetting. This effect is expected to be larger in setups where the total training set is large, many updates to the model pass before validation and inter institutional heterogeneity exists in the data. To combat this effect, the order of presenting batches to the central node can be interleaved, we call this mode *alternating-batches.* In this work we will consider the 'alternating-epochs method' the default because data in our experiments are homogeneous by default and because it simplifies the training procedure.

Spit Learning does not require centralization of data or labels but is functionally identical to training using centralized storage.[23] This solves aforementioned found in training in centralized fashion, while at the same time mitigating the cost of dropping inference performance compared to alternative DML methods.[24,81]

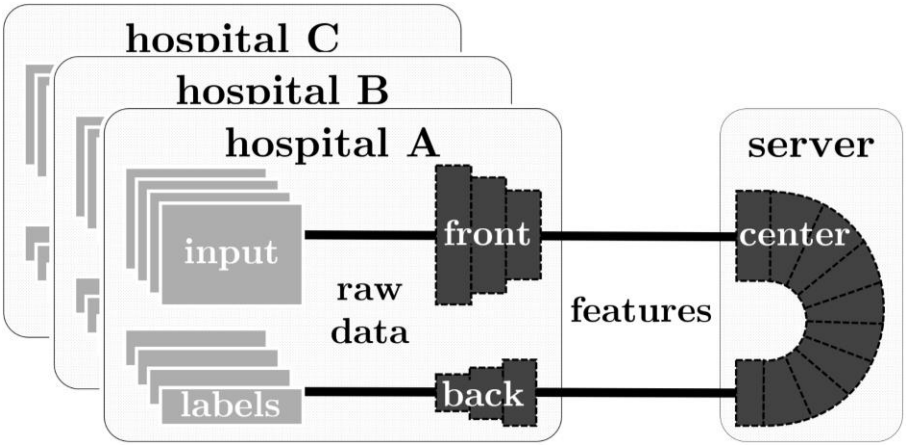



*Figure 5: Diagram of Boomerang Split Learning Three institutions named hospital A, B and C hold their own data and labels to collaboratively train a model without sharing raw data. The training process iterates over the hospitals of which hospital A is currently training.*

### **3.3.3** Performance aspects

Performance of SMPC methods in general can refer to many things. That is why in this work we refer to the inference error achieved as inference performance. In addition, we will be interested in training dynamics answering questions such as: Does training using Split Learning delay convergence? Therefor we define the convergence rate as the number of epochs required to achieve minimal loss on the validation set. To get a full grasp at feasibility of Split Learning we will also be interested in communicational and computational requirements.

### 3.3.4 Privacy and security

As mentioned earlier, most interest in SMPC goes out to protecting data from peers. Protection from outside attacks, like man-in-the-middle attacks, can already be assured conventional cryptographic methods like end-to-end-encryption.

Split Learning offers security by design through three properties. Firstly, commonly used activation and pooling layers are non-invertible, rendering the inversion operation a one-to-many problem. Secondly, the information sent over is a compact representation of source information making inversion a sparse problem. Lastly, recovery of underlying representations is further hindered as no single entity knows the full model state.[110]

Methods to provide additional security for Split Learning exist. Among these most prominent is the concept of *no-peek* Split Learning.[110] It proposes to decorrelate the features sent over from the local to a central node from the original raw data using a Kullback-Leibler (KL) divergence classifier. Methods like these can result in a drop of inference performance. No-peek Split Learning retains performance relatively well, it is a performance drop that can be avoided by creating a trusted eco-system if unreliable and adversarial or malicious participants can be avoided. We assume a trusted environment with transparent oversight of institution behavior to obviate methods to achieve this costly definition of security. [115]

As currently viewed, control and ownership of the ecosystem can remain fully with the participating institutions. Split Learning does allow online learning by default such that participating can be added or removed at any time. At any point in time, the model is optimized for all participating institutions. Currently, no methods exist to remove information gleaned from centrally trained models if institutions wish to quit participating, besides restarting training.

### 3.3.5 Other challenges

The distribution of data can present many challenges to methods for SMPC in general. For one, real world scenarios often present heterogeneous data. Inter-institutional heterogeneity can be caused by differences in hardware, scan protocols, sequences labeling protocols, or demographics and can deteriorate performance.[116] This is the case regardless whether training is performed in distributed or centralized fashion. This effect is generally greater in uncontrolled environments like clinical practice than the more controlled research environments. In current multi-center studies, data-preprocessing and common labeling protocols are employed minimize this effect.[81] In addition, data useful to make a predictions can reside at multiple institutions and be so called *vertically partitioned*. In this work, we will initially demonstrate the principle of Split Leaning on equally distributed, homogeneous, horizontally partitioned data. These concepts will be expanded on in chapter 5.

## 3.4 Conclusions

Multi-center machine learning can improve sample sizes and thus generalizability of predictive models. The hurdles that accompany medical data mentioned earlier demand for SMPC. Most methods do not scale well enough to support training of deep neural networks. The main sophisticated methods that do are Federated Learning, Cyclic Weight transfer and Split Learning. Split Learning splits a conventional neural network into sequential elements, of which a network split in three is called a Boomerang configuration. Only features are sent over, which are a product of many non-invertible operations, assuring security.

Split Learning is functionally more identical to training in centralized fashion than alternative methods. In addition, the modularity provided by Split Learning can reduce computational requirements, lowering the entry bar for participation. The communication costs are totally different and highly domain dependent. Lastly, heterogeneous data can form a challenge for SMPC methods. The effect of these factors has not been researched for medical use cases.

# 4 Split Learning Feasibility

In this chapter we will investigate the performance aspects of the default Split Learning, its applicability to health care, and how these results match up against alternative distributed as well as the centralized approach. To achieve this, this work presents the first application of Split Learning in literature and provides several adaptation and insights for future implementation. Implementations of four clinical tasks are described. Their performance is analyzed in four domains: inference performance, convergence rate, computational requirements, and communication overhead. Based on our findings we present a verdict on Split Learnings applicability compared to alternative methods, as well as adaptations relevant to Split Learning enabled medical deep learning.

## 4.1 Aim

In this chapter we assess Split Learning feasibility for medical imaging applications. We split feasibility into three topics:

i. Applicability: Can we apply popular neural networks to Split Learning, and what adaptations would have to be made? Secondly, even if it is possible, what are the limitations imposed by Split Learning on the flexibility of these implementations.
ii. Performance: How does training of models using Split Learning compare to models trained using models trained in centralized fashion?
iii. Requirements: Is there a workable overhead due to communication. And how are institution-side computational requirements reduced by enabling outsourcing?

## 4.2 Methods

### 4.2.1 Split Learning implementations

An in-house built virtual Split Learning testbed was built for simulation of variable Split Learning setups to assess inference performance, convergence rate, memory cost and bandwidth cost. The test bed was built in Python (v3.7.5) and PyTorch (v1.2.0) on both Windows (Windows 10) and Ubuntu (v18.04.3 LTS) and is made openly available.[117] Core functionality of the testbed was implemented based on algorithm 1 provided in appendix 8.2. The testbed provided modularity to support quick implementation of several medical applications. Four representative non-trivial medical imaging tasks were chosen to represent a variety of tasks, data, sample sizes and models.

Data was partitioned into training and validation partitions, used during the training phase, and an independent test set to compute inference performance values on. The training and validation sets were distributed equally over all participating institutions, but final performance computation was performed on the entire test set. The next part describes the data and task briefly, it goes in depth on the networks applied to Split Learning. A summary of these implementation can be found in Table 1. An in-depth description of each data set accompanied by visualizations of the networks described below can be found in the appendix at 8.1.

**Diabetic Retinopathy Challenge (DRC)**: Diabetic Retinopathy is a complication of diabetes that affects the blood vessels in the retina and forms the leading loss of blindness[42]. However, 90% of cases can be reduced by proper treatment and monitoring of the eyes after diagnosis[118]. Diagnosis usually consists of acquiring fundus photography images. A sample fundus photograph is provided in Figure 6. Fundus photos from the Diabetic Retinopathy Challenge (DRC)[119] were classified for being normal or abnormal (presenting a stage of diabetic retinopathy) using a 34-layer residual network[120] (ResNet34).

*Figure 6: Example fundus photograph from the DRC data set used to classify if diabetic retinopathy is present.*

Residual networks are inspired by pyramidal cells in the cerebral cortex. They contain *skip connections* that connect two non-sequential layers, bypassing others. This overcomes a fundamental problem in neural network optimization of vanishing gradients, thus allowing for deeper networks. To adapt this architecture to Split Learning is simplest when split in between these skip connections. Residual networks contain comparatively many parameters, which play a role in communication equations for distributed machine learning. Because security is dependent on the non-invertible operations in the frontal link, this link must be sufficiently deep. Because the final layer is a fully connected layer, and information presented in this fully connected layer is highly randomized compared to final classification results, depth of the back link does not require this depth, improving institution-side computational load. Inference performance was measured on an independent test set and defined by the accuracy of the classification.

**Brain Tumor Segmentation (BraTS):** Accurate segmentation of high grade glioblastoma is crucial in monitoring tumor growth dynamics, surgical results, or tumor response after oncological treatment.[121] This segmentation task was implemented using fluid attenuated inversion recovery (FLAIR) magnetic resonance imaging (MRI) from the BraTS challenge data[122–124] set using a 3D U-Net[125] architecture. A sample FLAIR image is provided in Figure 7. Since it was introduced in 2015, the U-Net has become one of the most popular topologies for biomedical image segmentation. The network consists of an auto-encoder: a contracting path followed by an expanding path, with

a bottleneck layer in between. In addition, every feature map in the contracted path is concatenated with the corresponding feature map in the expanding path, forming *skip connections*. To ensure security in the adaptation of this architecture to Split Learning, high level features sent through skip connections, originating from local contracting layers, should not be concatenated with expanding layers residing in central layers. We propose and implemented a Split Learning adaptation in which the architecture is split symmetrically around the bottleneck layer with the exterior links local and center link central, visualized in Figure 17. To reduce the local computational burden, and to improve the security, the network is split after the second contracting layer. This setup is used to segment high grade glioblastoma from fluid attenuated inversion recovery (FLAIR) magnetic resonance imaging (MRI). Segmentation performance is quantified using the Sørensen–Dice coefficient on an independent test set.

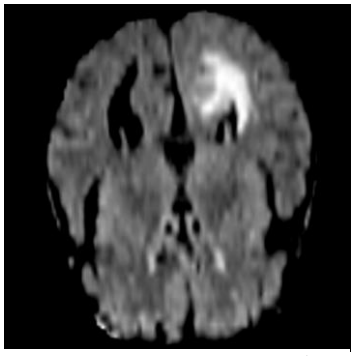


*Figure 7: Example FLAIR MRI from the BraTS data set used for tumor segmentation.*

**Chest X-ray multi label classification (CheXpert):** Chest radiography is the most common imaging examination globally. It is critical for screening, diagnosis, and management of many life-threatening diseases. Automated interpretation could improve workflow prioritization and clinical decision support. The Large Chest X-ray data set CheXpert[126] presents a multi-label classification task, classifying the presence of fourteen classes. It was implemented using a 121-layer dense network (DenseNet)[127]. The network consists of a series of *dense blocks* followed by *transition layers*. The DenseNet is popular because of its low number of parameters and higher performance on low number of data samples. This is achieved by the dense blocks containing *dense connections* that propagate superficial input much deeper into the network. This also means these dense blocks are hard to split for Split Learning. We therefor propose to split the network after the first transition layer. Input is a chest X-ray image, and output consists of a vector of fourteen binary values predicting the presence of the fourteen observations. This architecture is visualized in Figure 18. Implementation as described according to CheXNet[128] was used. Inference performance on the independent test set defined according to the method described in the original paper as arithmetic mean area under the receiver operating characteristic (AUROC) over five competing tasks (cardiomegaly, consolidation, atelectasis, pleural Effusion and edema).

*Figure 8: Example Chest X-ray sample from the CheXpert data set from which presence of several of fourteen findings are to be established.*

**Musculoskeletal Radiographs (MURA):** Musculoskeletal conditions are the most common cause of severe, long-term pain and disability.[129] Abnormality detection in musculoskeletal radiographs allows for worklist prioritization and com-bat radiologist fatigue. MURA is a large data set containing musculoskeletal radiographs labeled for presenting abnormality by radiologists presenting a binary classification problem. Instead of a 169-layer dense network that is most common used, we used a 152-layer residual network (ResNet152) to include a model with more parameters in our comparison. The adaptation made and its reasoning are the same for this network as for the DRC described above.

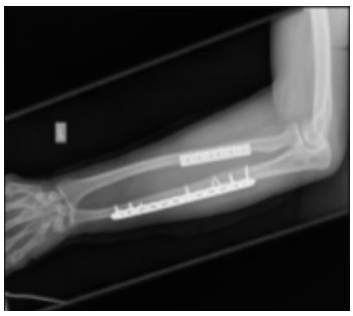


*Figure 9: Example of an elbow radiograph from the MURA data set used for abnormality classification.*

These implementations are summarized in Table 1. An in-depth description of each data set accompanied by visualizations of the networks described below can be found in appendix 8.1.

A summary of the implemented tasks is provided in Table 1. The splits of conventional network $F$ determines properties of the resulting links $F_{front}, F_{center}$ and $F_{back}$, such as number of parameters $N$ and size of the interface layers.

*Table 1: Summary of implemented medical imaging tasks.*

| Data set | Image type | Task | Topology | Measure |
|---|---|---|---|---|
| **DRC**[119] | Fundus photographs | Binary Classification | ResNet34[120] | Accuracy |
| **BraTS**[122–124] | FLAIR MRI | Binary Segmentation | U-Net[125] | Dice |
| **CheXpert**[126] | Chest X-ray | Multi-label Classification | DenseNet121[127] | AUROC |
| **MURA**[129] | Musculoskeletal Radiographs | Binary Classification | ResNet152 | Accuracy |

### 4.2.2 Validation of the testbed

The tasks above were implemented in a virtual Split Learning testbed. Each of the implementations was validated by training them in the Split Learning testbed in centralized fashion and comparing their inference performance to values found in literature.

The virtual testbed provided a functionally identical Split Learning environment for performance research presented in this work. In parallel networking code developed and made openly available by our collaborator Kevin Pho.[130] The scope of this work was therefore limited to virtual performance analysis. However, to affirm the validity of the testbed it was finally validated using a MNIST classification task where one institution was situated at the Advanced X-ray Imaging Sciences Center in the department of Radiology at the Massachusetts General Hospital running ArchLinux (v2019.11.01), and another at the Applied Chest Imaging Laboratory at Brigham and Women's Hospital running Ubuntu (v14.04.6 LST). The relay server was located at the Massachusetts Institute of Technology Media Lab cluster running Ubuntu (v18.04.3 LTS). In this

validation internet speeds and local computational requirements were measured to compare to be used in communication and computation cost calculations.

### 4.2.3 Inference performance and convergence rate analysis

We investigate if inference performance and convergence rate are lower compared to training on centralized data by computing Wilcoxon signed-rank test between repeated measurements of centralized training and Split Learning where the number of participating institutions is two. Secondly, we ascertain if inference performance and convergence rate in Split Learning are affected by an increasing number of participating institutions by computing the same test for Split Learning for two, and Split Learning for fifty participating institutions.

To establish the effect of an increasing number of participating institutions $K$, many Split Learning models are trained using identical random initialization for $K$ ranging from 1 to 100 institutions. The relation between performance and $K$ without collaboration was measured to be closely linear related to $\log(K)$ ($\rho = -0.98$) Therefore, Pearson's rank correlation is computed between the $\log(K)$ and the inference performance, and $\log(K)$ and the epoch of minimal validation loss.

Finally, as an addition performance is also given for the scenario where the total data set is split, but institutions are not collaborating. When compared to Split Learning, these results provide insight in the added benefit of collaborative multi-center training. We expect any these results to be hard to translate quantitatively to other real-world use cases. Therefore, these experiments are of lower priority than other experiments and are only performed for the CheXpert and DRC challenges.

### 4.2.4 Computational and bandwidth cost analysis

Computational requirement was defined as the minimum amount of GPU memory required to run at a batch size of 32. In combination with the fraction of networks residing locally this results in an estimation of local computing requirements. This estimation was further validated by the earlier multi-machine validation process.

Relative communication cost of Split Learning to Federated Learning depends on size of the local models, size of the interface layers and total size of the data set. They follow equations proposed by Singh et al.[131]. These equations were adjusted to account for the extra split layer after the central link that is present in Boomerang Split Learning. Communication cost in Split Learning is dependent on: number of participating institutions $K$, total number of model parameters $N$ which is a sum of $N_{front}$, $N_{center}$ and $N_{back}$, total data sets size $p$ in bytes, total size of transmitted layers $q$, a sum of $q_{front}$ and $q_{back}$ in bytes, fraction of total number of model parameters residing locally given by $\eta = \frac{N_{front}+N_{back}}{N}$, communicational overhead $\Omega$, computation time per batch $\tau$ and finally internet speed $v$. For comparison, communication cost of Split Learning relative to Federated Learning is expressed following the equation:

1) $$\phi = \frac{pq}{NK} + \frac{\eta}{2}$$

We apply this equation to each of our implementations using found values of $N$ and $\eta$. Number of samples and number of institutions for each data set was $10^4$ and 10 for DRC, 250 and 19 for BraTS, $10^5$ and 20 for CheXpert and $4{\times}10^4$ and 5 for MURA.

Communicational overhead is defined as the increase in computation time compared to a centralized approach. This overhead depends on computation time per batch $\tau$ in seconds, $q$ in bits and communication bandwidth speed $v$ in bits per second, and is estimated by:

*2)* $$\Omega = \frac{q}{v\tau}$$

Note that, when $\Omega < 1$, time required to send a batch of features (communication time) is lower than the time required to perform computation on those features (computation time). When performed efficiently in parallel, this means that the communication required for Split Learning could be performed entirely during the computation of the previous sample-pair. This would negate overhead due to communication.

Computation time per batch $\tau$ is measured for each of the different Split Learning implementations on several consumer grade processing units. We used Nvidia GeForce GTX 1080 Ti, 2080 Ti TITAN X and Tesla P100 graphics processing units (GPU) and made sure training was performed under constant maximum GPU load by increasing number of workers. Internet bandwidth was measured over ethernet at the Massachusetts General Hospital and assumed fixed at $v = 694$ Mbps which is representative for modern fiber internet.[132]

## 4.3 Results

### 4.3.1 Resulting topological properties

The properties of the links resulting from the proposed splits are summed up in Table 2: Tasks and implementations summaries.

*Table 2: Tasks and implementations summaries. Number of parameters N, percentage of parameters that resides locally $\eta$ and size of the interface layers q*

| Data set | $N_{front}$ | $N_{center}$ | $N_{back}$ | $q$ | $\eta$ |
|---|---|---|---|---|---|
| **DRC** | 9.5k | 21.3M | 513 | 756.4k | 2.40% |
| **BraTS** | 3.2k | 2.0M | 17.5k | 580k | 1.00% |
| **CheXpert** | 1.7M | 5.2M | 16.4k | 501.8k | 24.63% |
| **MURA** | 9.5k | 60.2M | 513 | 756.8 | 0.87% |

### 4.3.2 Validation of the testbed

Single institution DRC models achieved on average 78.3% accuracy, compared to 78.7% found in literature.[113] Sørensen–Dice-Scores achieved on the BraTS set were 0.851 compared to 0.862 found in literature.[81] Single performance for CheXpert achieved an AUROC of 0.866 compared to 0.855 and 0.889 in the original CheXNet[128] and CheXpert[126]papers respectively. The accuracy

performance metric for MURA did not match results found in literature such that validation was not possible.

### 4.3.3 Inference performance and convergence rage

Results on inference individual inference performance and convergence rate measurements are visualized in Figure 10 and Figure 11 respectively. Mean values for these results and their correlation to $K$ are given in Table 3.

*Table 3: Results of number of participating institutions on performance and convergence.*

| | inference performance | | convergence rate | |
|---|---|---|---|---|
| Data set | $\mu \pm 2\sigma$ | $\rho$ | $\mu \pm 2\sigma$ | $\rho$ |
| **DRC**[119] | 78.3±0.5% | 0.110 | 119±19 | 0.274 |
| **BraTS** | 0.851±.008 | 0.113 | 217±68 | 0.037 |
| **CheXpert**[126] | 0.866±.003 | -0.099 | 2.3±0.4 | 0.011 |
| **MURA** | 77.2±1.4% | -0.141 | 5.5±1.0 | 0.006 |

Mean classification accuracy on the DRC task was 78.3% ± 0.5% (mean ± 95% C.I.). Segmentation accuracy on the BraTS task scored a Sørensen–Dice coefficient of 0.851 ± 0.008. The CheXpert task achieved an AUROC of 0.866 ± 0.004 and the MURA task reached an accuracy of 77.2% ± 1.4%. Correlation to number of participating institutions $K$ were $\rho = 0.110$ for the DRC, $\rho = 0.113$ for the BraTS, $\rho =$ -0.099 for the CheXpert and finally -0.141 for the MURA data set.

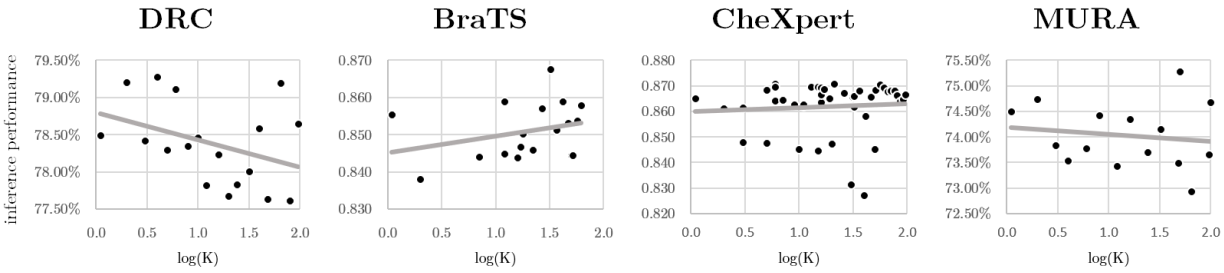



*Figure 10: Scatterplot of inference performance* over log(K) for each implemented task with linear trendlines.

The epoch of optimal performance was 199 ± 19 for the DRC, 217 ± 68 for the BraTS, 2.3±0.4 for the CheXpert and 5.5±1.0 for the MURA data set. Correlation to number of participating institutions $K$ were $\rho =$ -0.274 for the DRC, $\rho = 0.037$ for the BraTS, $\rho = 0.011$ for the CheXpert and finally 0.006 for the MURA data set.

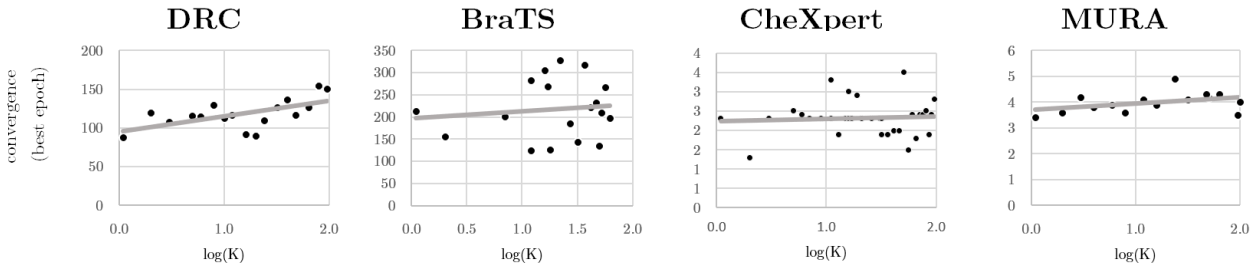



*Figure 11: Scatterplots of convergence rates over $log(K)$ for each implemented task with linear trendlines.*

Performance results for non-collaborative settings compared to Split Learning are provided in Figure 12.

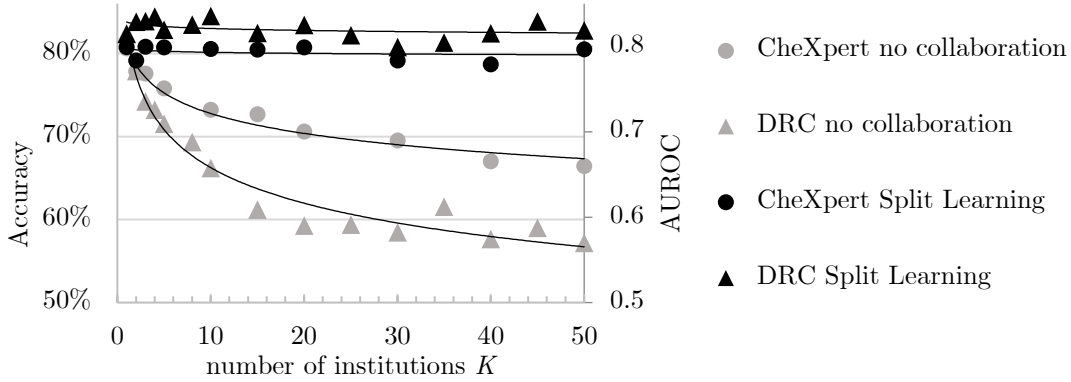



*Figure 12: The performance gain of collaboration. When a constant amount of data is split of a number of participating institutions inference performance drops steeply when not collaborating while remaining constant when using Split Learning.*

### 4.3.4 Computational and bandwidth cost results

Computational requirements are described and communicational requirements both as compared to federated learning and as overhead are provided in Table 4.

*Table 4: Results on computational and communicational requirements.*

| Data set | computational requirement | Communication *relative to FL* | *overhead* |
|---|---|---|---|
| **DRC**[119] | 2.9 GB | 34.7 | 0.035 |
| **BraTS** | 33.4 GB | 3.3 | 0.007 |
| **CheXpert**[126] | 11.7 GB | 359.3 | 0.008 |
| **MURA** | 26.3 GB | 125.6 | 0.007 |

## 4.4 Discussion

Our results also show low correlation between increasing number of institutions and performance, indicating functional similarity between training with centralized data and Split Learning. We also conclude that Split Learning is functionally scalable for a realistic range of clinical participants. Our results clearly demonstrate the performance benefit can be achieved by pulling data resources together, regardless of network type, task or data set size.

Alternative methods like Federated Learning suffer from performance loss because of their fundamentally different aggregation-based training protocol. This effect is a larger benefit when single institutions hold small data sets, which is often the case in medical multi-center collaboration. This difference also affects the comparative computational load which can be up to four times higher in Federated Learning compared to Split Learning. In practice this means that hospitals would not have to possess state of the art hardware to participate in multi-center deep learning. However, these benefits come at the loss of bandwidth efficiency. Many Split Learning implementations with common health care parameters using conventional architectures are expected to require higher bandwidth usage than Federated Learning. Split Learning is more bandwidth efficient in use cases when the total amount of data is small, many institutions are participating, and models are large. However, we do not expect this bandwidth to form a problem for hospitals with modern access to internet. Further research into the practical implications of bandwidth usage is necessary to confirm these estimations. Alternatively, Split Learning bandwidth usage could be improved by compressing intermediate representations or reduction of feature dimensions for example using auto encoder-based compression.

Our work demonstrates feasibility of Split Learning for medical imaging problems by showing application on different representative tasks. Although our results are applicable to a range of similar problems, strong variation on the parameters chosen could for example change the logistical requirements. Variations within the medical domain include Split Learning for non-imaging problems and training in resource constrained environments. Resource-wise, imaging is relatively demanding for Split Learning, which is why we expect our results to also extend to non-imaging problems. In resource constrained situations, for example for rural hospitals with limited bandwidth, the communicational overhead could increase drastically. However, the benefit of centralization of computing could be more important and remains.

Split Learning relies on intermediate representations generated from several layers of non-invertible operations being sent between local and central entities. In certain situations where the system contains adversarial institutions, caveats could be found in the notion of security this appeals to. Although further research is being performed aiming at more rigorous definitions of security, they currently come at a cost in performance. In health care we propose aiming at creating a trusted environment before implementing such methods.

In practice, data sets are seldomly as homogeneous as presented in this proof of concept, which poses a challenge for machine learning in general. This usually requires collective systematic data preprocessing, which would be of no difference from preprocessing in a centralized setup. However, recent research has stressed the increased effect of heterogeneous data in distributed compared to centralized training.[116] We believe opportunities to lie in the modularity of Split Learning to provide a basis for a solution for heterogeneous data that we will touch upon in the next chapter.

Institutions can be added or removed from participation at any time. This further increases the level of control that remains with participating institutions. Currently, no method exists to remove information specifically supplied by leaving institutions, requiring retraining of such models. Solutions could lie in performing stochastic gradient ascent to find alternative local minima for the new set of collaborating institutions.

Future implementation of Split Learning systems could be embedded in existing AI-platforms like Nvidia Clara, General Electronics (GE) Edison or the American College of Radiology (ACR) AI Lab. Distributed machine learning infrastructure in the process of being embedded for Federated Learning could be leveraged to speed up deployment of Split Learning.

## 4.5 Conclusion

Privacy preserving distributed deep learning can enable health care institutions to collaborate on training deep learning models.

Our results show that Split Learning is feasible for medical imaging applications and presents several opportunities on each aspect of feasibility we analyzed:

i. Applicability: Implementation of all common networks was possible, but presence of skip connections must be taken into account to prevent sharing of high-level features with the centralized node.
ii. Performance: In contrast to many alternative distributed machine learning paradigms, inference performance and convergence rate achieved using Split Learning do not seem to be affected by an increased number of institutions. Because Split Learning is functionally identical to conventional training this also implies no performance is lost compared to centralized approaches.
iii. Requirements: Split Learning bandwidth consumption is significantly higher than Federated Learning in most cases but does not increase overall training time with modern internet access. Institution-side computational resources requirements can be significantly lowered using Split Learning, enabling institutions to participate without investment in hardware.

Our results affirm suitability of Split Learning for medical imaging applications and confirm favorable properties compared to both alternative distributed as centralized approaches.

Future challenges for Split Learning lie in translating these results from sterile hypothetical challenges, to be more robust to be able to face real world challenges. For security, this means creating a more rigorous description of an environment that would obviate the need of additional security methods, like no-peek Split Learning. For handling of real-world data, further research is required to ascertain the effect of, and come up with solutions for inter-institutional heterogeneity.

# 5 Split Learning Innovation

Previous chapter investigated feasibility and appeal of Split Learning for medical imaging applications. But beyond the previously presented '*vanilla*' use case for Split Learning many alternatives can be imagined that present novel opportunities but also come with specific challenges. In this chapter, we focus on three fundamental challenges. These are training on heterogeneous data, increasing security and handling of '*vertically partitioned*' data streams.

The Split Learning paradigm involves modularity which we employ to propose solutions to these challenges. We propose an alternative training scheme to allow for inter-institutional domain adaptation by design, we experiment with alternative weight-sharing-strategies to improve security and communication cost, and lastly, we discuss opportunities in combining data streams from multiple institutions.

## 5.1 Aim

In the previous chapter Boomerang Split Learning has been introduced. From now on we will call this '*vanilla*' Split Learning. This chapter will discuss and evaluate three different adaptations to it disjointly. Their method and purpose are as follows:

i. Local Adapter Networks to solve inter-institutional heterogeneity
ii. Alternative weight-sharing protocols to improve security
iii. Vertical Split Learning to handle vertically partitioned data

### 5.1.1 Local Adapter Networks to solve inter-institutional heterogeneity

One assumption underlying deep learning models –one we implicitly employed in the previous chapter– is that training and test data are independent and identically distributed (IID). This means that the draw of data points does not influence the outcome of subsequent draws and that the distribution does not change at some point.[133] Non-IID data is a challenge for machine learning in general. For example, when data trained on does not generalize during inference for testing data, or -even worse- during deployment. But for distributed machine learning, it also poses a fundamental challenge during the training phase.[116] In federated learning for example, aggregation of the gradients can see subtractive averaging, decreasing performance.[22,24] Other methods like Cyclic Weight Transfer could suffer from catastrophic forgetting as the optimization processes aim to reach different minima.

Many causes exist for non-IID data. However, in this part we specifically focus on the challenges that arise with multi-center data, being inter-institutional differently distributed data sets. In chapter 2 it was described that generation of medical data is a complex process, in which each step can introduce variation that violate the IID assumption. This variation can be of epidemiological, demographical or acquisitional origin. These epidemiological differences in prevalence are also known as *label imbalances.* Differences in demographics can cause non-representative populations. One example of skewed demographics is the Veterans administration.[79] And lastly, variation in acquisition can cause a difference of the data distribution within the same task. This is called *data set bias* or *domain shift.* Domain shift can be caused by the wide variety of acquisition protocols, hardware and contextual environments that make up the acquisition process.

No quantitative research exists on the effect of clinical sources of heterogeneity on distributed machine learning. This chapter will focus on domain shift, as the other challenges are only a major challenge in specific situations and can be solved by more conventional solutions such as strict inclusion and labeling protocols.

Practical sources of inter-institutional domain shift can be differences in equipment or inclusion and acquisition methods and protocols, an example of which is visualized in Figure 13. Domain shift is usually difficult to define formally although domain shift as simple as differences in for example pixel intensity distributions have shown to be enough to impede generalization of trained models to other data sets,[6,134]

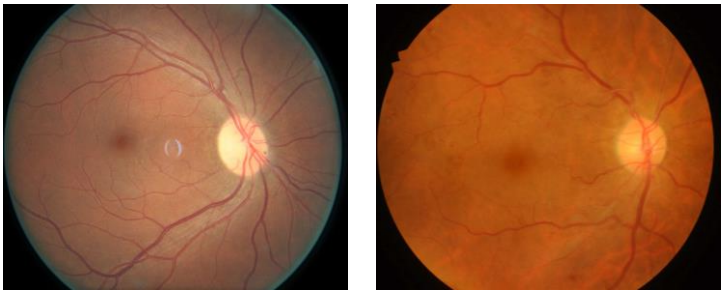


*Figure 13: Example of domain shift: Two semantically similar images from different scanners.*

For training, the solution is to devise a transformation to convert any domain specific raw data into domain independent representations. In current multi-center studies this transformation often takes place in the preprocessing stage, using a manually designed set of operations that have been previously observed to be useful, like high-pass filtering[135] and cropping. However, configuring an optimal transformation can be challenging. This is especially true access to the distributed data sets is limited. In addition, there is no guarantee that the manually configured preprocessing steps produce optimal domain independency of the representations.

The modularity of Split Learning allows for novel solutions to tackle this problem, potentially further lowering the entry barrier for participation and improving performance. The method we

propose employs the front in a Boomerang configuration as an adapter netowork[134] to perform institution specific domain adaptation. In this subsection we initially demonstrate that inter-institutional variation can affect performance, and subsequently demonstrate effectiveness of proposed solution. Finally, we test the generalizability of this method on inter-institutional MRI heterogeneity due to variation in sequencing.

### 5.1.2 Alternative weight-sharing protocols to improve security

In the previous chapter we discussed how SMPC assures security from its peers by protecting information about raw input and output data. In Split Learning participating institutions only require the local parts of the network, and the server only requires the central part.[110] This provides an extra layer of security as the full architecture of the network is not required to be known by any single contributor.

In the previous chapter we assumed the environment to be secure and behavior of clients to benign. However, in less trusted environments these systems are vulnerable to membership inference, or adversarial example attacks because an attacker could recreate full model internals of another institution. Such an attacker would be able to do so because it shares the exact local nodes with each other institution and would be able to recreate the internals of the central model.

For such environments, security could be further improved by an alternative weight sharing strategy. In this chapter we tested with not sharing certain local links, such that they would never be sent over. This adaptation is expected to cause performance loss, or even instability as each institution's links are only trained on a subset of the total amount of available data. In this subchapter we investigate the price of performance of these weight sharing strategies.

### 5.1.3 Vertical Split Learning to handle vertically partitioned data

The most common reason for medical multi-center research is to increase the number included number of data points, such as patients or sessions. This has implicitly been the type of multi-center collaboration that has been discussed throughout this work until now. In this case, for every unique data point all features that can be used for prediction reside with a single institution and increasing the number institutions improves prediction because the combination of these data subsets presents a more complete representation of the probability distribution. According to this concept the entire data pool is split *horizontally*. An example of a horizontally partitioned data set is given in Table 5.

*Table 5: Example of features (F) of several patients split horizontally. This is the case for most multi-center studies.*

| Data point key name | *F1: age* | *F2: BP* | Data point key name | *F3: LDL* | *Outcome: mortality* |
|---|---|---|---|---|---|
| **Alice** | 64 | 125/83 | **Alice** | 180 | 51 |
| **Bob** | 72 | 140/81 | **Bob** | 130 | 20 |

| Data point key name | *F1: age* | *F2: BP* | Data point key name | *F3: LDL* | *Outcome: mortality* |
|---|---|---|---|---|---|
| **Carol** | 81 | 142/93 | **Carol** | 117 | 73 |
| **Dave** | 58 | 123/79 | **Dave** | 159 | 13 |

Alternatively, there is another method of multi-center collaboration. It is possible to improve prediction accuracy by acquiring not more data points, but by acquiring more features of a data point. In this concept all institutions possess different features of every data point that could together be used to cast a prediction. Such a data set is said to be partitioned *vertically*. An example of a vertically partitioned data set is provided in Table 6.

*Table 6: Example of features (F) of several patients split vertically. This notion of partitioning is less common for medical data.*

| Data point key name | *F1: age* | *F2: BP* | Data point key name | *F3: LDL* | *Outcome: mortality* |
|---|---|---|---|---|---|
| **Alice** | 64 | 125/83 | **Alice** | 180 | 51 |
| **Bob** | 72 | 140/81 | **Bob** | 130 | 20 |
| **Carol** | 81 | 142/93 | **Carol** | 117 | 73 |
| **Dave** | 58 | 123/79 | **Dave** | 159 | 13 |

There are two reasons why horizontally partitioned data has been the most common notion of multi-center data. For one, horizontally-partitioned multi-center research is just a scaled-up version of single-institutional research, thus the same protocol is feasible without collaboration and it is thus logistically simpler to conceive. Secondly, standardized data did not use to exist in the abundance as it does today. Therefore, the medical diagnostic process is not yet geared towards using externally generated data. However, a large part of the explosion of data produced is vertically partitioned. This includes for example data from mobile devices, activity trackers and other internet of things devices. It requires an integrated solution to make use of this data in a secure manner.

Redefining medical data to also include data not acquired inside the hospital presents great opportunities which have mostly been investigated with respect to new diagnostic methods for mental health as of yet.[136–138] These benefits stem from the relieving the constriction that medical data is acquired only sporadically, in a limited amount of time. As an example, mobile phone app usage can already distinguish healthy from symptomatic cognitive impaired patients.[139] Presenting a more continuous diagnostic method.

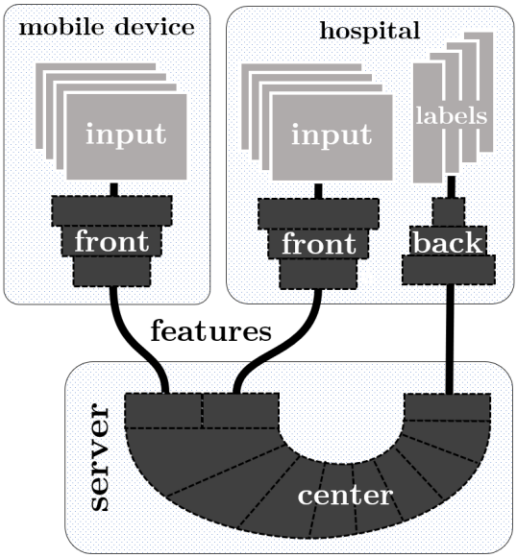


*Figure 14: Diagram of data flow in Split Learning for vertically partitioned data*

However, employing vertically partitioned data also requires vastly new methods for handling these data streams. In contrast to alternative distributed machine learning methods, Split Learning allows for seamless integration of externally acquired data alongside other data streams like in-house acquired data. In its most simple form, such a configuration is provided in Figure 14. Passing this technical hurdle would open up opportunity for researchers to start working at scale on some many unique ethical and clinical challenges that this form of data poses to be useful for health care applications.

Utilization of such externally generated data has not yet seen medical applications as of yet. Therefore, it lies beyond the scope of this work to implement and testing such systems but encourage further work to do so.

## 5.2 Methods

### 5.2.1 Split Learning with Local Adapters

We first confirm the negative impact of inter-institutional heterogeneous training data in a worst-case scenario using synthetically transformed data. In response, we propose a solution to this class of problems we name Split Learning *with local adapters*. These two steps form a proof of concept providing quantification of the problem as well a solution. Secondly, we test if this problem and its solution generalize for real world heterogeneity.

#### *5.2.1.1 Proof of concept*

Our initial experiment is performed on a setup based on the DRC configuration detailed on in previous chapter, with four participating institutions. Initially a model is trained using the default vanilla Split Learning protocol, using default IID data and baseline performance is computed on an independent test set. The DRC is a binary classification problem with exactly matched classes such that random chance would result in a 50% accuracy.

Secondly, we a apply an institution specific transformation to the data and perform exactly the same experiment. The transformation is designed to separate every institutions data set into non-overlapping distributions. The transformation used were different normalization boundaries being [0, $2^8$], [-2 $\sigma$ , 2 $\sigma$ ], [0, $2^{12}$] and [-10, 10]. Although trivial to correct using normalization in preprocessing, they are the most basic example of data heterogeneity.

Thirdly we propose a solution to the challenge above we call Split Learning with Local Adapters. The core concept of this method revolves around allowing each institution to train its front link as a proprietary adapter to convert its data to domain independent representations. This theory is based on AdapterNets for domain adaptation.[134] Each institution does this in an additional training stage before the full chain of links is trained. In this stage, the center and back are frozen such that in training their weights are not updated, and only the front adapter is updated until it converges. This provides an unsupervised method that is completely explainable by looking at the domain independent features sent over. The adjustments of this algorithm to default Split Learning is provided in Algorithm 2 in appendix 8.3.

As a last tweak to this approach it still leaves two options. In its most basic form, states of front links could be chosen not to be shared as each institution would train a different adapter anyways. In this configuration front links state would remain mostly constant. Alternatively, front links states could be chosen to be forwarded just like in regular Split Learning. This could be explained as transfer learning one adapter from another. We compute performance for both methods and compare achieved accuracy to vanilla Split Learning.

#### *5.2.1.2 Real world data*

To reinforce our conclusions, we repeat the same experiment using institutions using real MRI data. The T2 and FLAIR sequences were chosen from the BraTS data set as they display edema caused by the tumor by the same physical properties but the images on a whole display vastly different domains. These scans are visualized in Figure 15. Additionally, mutual information in these sequences has already been demonstrated by means of style transfer using generative adversarial networks.[140,141]

To repeat the previous experiment, initially two baseline performances are computed: one where every one of the four institutions hold 25% of all FLAIR scans, and another similarly for the T2 scans from the BraTS data set. Secondly, heterogeneity is introduced by providing two institutions with 25% of the FLAIR and two with 25% of the T2 data sets so that no scans from one patient overlap the groups but all remain present. In this heterogeneous setting vanilla Split Learning, Split Learning using Local Adapters, and Split Learning using local adapters with transfer learning are tested. The architecture of the adaptors consists of several convolutional blocks as proposed in the original paper.[134]

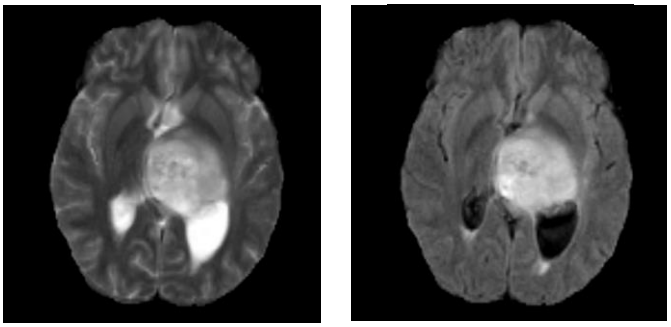


*Figure 15: Example T2 (left) and FLAIR (right) MRI scans presenting domain shift. Visualization of glioblastoma in the T2 is based on the same physical properties as the FLAIR but the images present a domain shift that is hard to correct using conventional preprocessing methods.*

### 5.2.2 Variety of weight sharing strategies.

In the Boomerang configuration introduced earlier the options of weights sharing among institutions that we will experiment with are:

i. full weight sharing (as used in the previous chapter)
ii. sharing back link only
iii. sharing no links at all

One option we did not investigate was 'sharing front link only'. Because of the high complexity of the fully connected layer in the back link and its expected large effect on achieved performance, in combination with it being farter form interpretable data it was expected that the performance cost would severely outweigh potential benefits.

To provide a baseline of the performance that Split Learning still achieves even when no links shared, non-collaborative performance is also computed. All of these strategies are computed for a number of institutions ranging from 1 to 50 using the DRC data set.

## 5.3 Results

### 5.3.1 Split Learning with Local Adapters

Proof of conceptTable 7 presents performance as classification accuracy of models trained for the proof of concept using trivial transformations on the DRC data with 50% accuracy being random chance.

*Table 7: Inference performance on trivial non-homogeneous data.*

| Homogeneous data | Training Protocol | Accuracy |
|---|---|---|
| | Vanilla Split Learning | 79.15% |

| Heterogeneous data | Training Protocol | Accuracy |
|---|---|---|
| | Vanilla Split Learning | 52.96% |
| | Local Adapters | 75.37% |
| | Local Adapters + Transfer | 80.78% |

#### *5.3.1.1 Real world data*

Table 8 presents performance as Sørensen–Dice segmentation quality on real world data consisting of FLAIR and T2 MRI scans.

*Table 8: Inference performance on real non-homogeneous data.*

| Homogeneous data | Training Protocol | Dice |
|---|---|---|
| T2 | Vanilla Split Learning | 0.809 |
| FLAIR | Vanilla Split Learning | 0.849 |

| Heterogeneous data | | Dice |
|---|---|---|
| T2/FLAIR | Vanilla Split Learning | 0.621 |
| T2/FLAIR | Local Adapters | 0.691 |
| T2/FLAIR | Local Adapters + Transfer | 0.724 |

### 5.3.2 Weight sharing options

Performance for different number of participating clients along with logarithmic trend lines is provided in Figure 16.

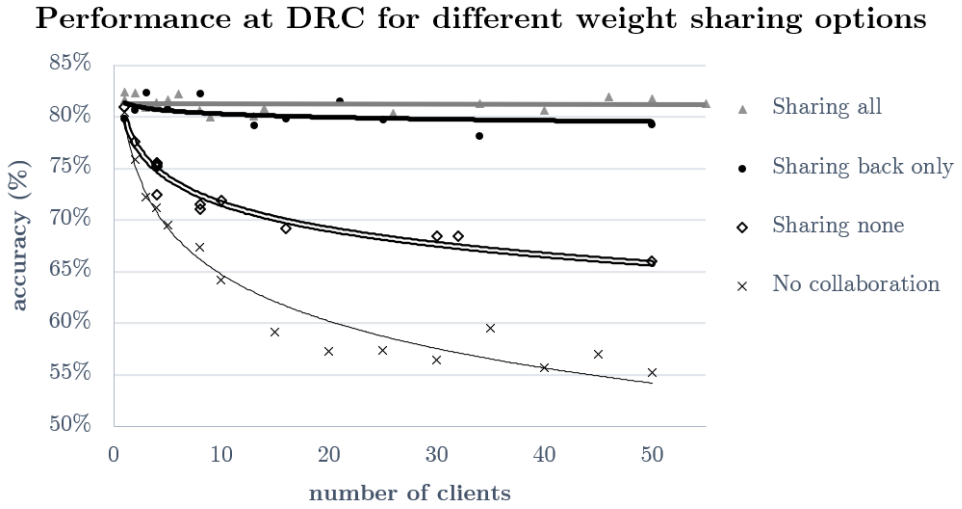



*Figure 16: Performance for different weight sharing options.*

## 5.4 Discussion

In order to achieve successful implementation of Split Learning for real world applications several challenges not represented in the scenario presented in the previous chapter would require investigation. For distributed machine learning in general these include but are not limited to heterogeneous data, security for applications in less secure environments and handling of non-horizontal data streams.

Our results with respect to the first topic, show that heterogeneous data can have severely detrimental effects on Split Learning performance. The expected cause is each institution to have individual very different minima that no single model configuration can achieve proper optimization for. We propose a solution that employs the front link of the Split Learning chain as adapter Networks to transform raw data suffering from domain shift to domain independent representations that the subsequent model can optimize for properly. Our initial results show promising recovery from synthetically induced transformations. Recovery from heterogeneity is less striking when tested on real world data but still apparent. This might be due to the relative simplicity of the adapter to the complex domain shift present. Results from the method utilizing transfer learning are higher than without, reinforcing our believe that transfer learning among similar adapters provides a performance benefit. To investigate the full potential of this method more capable adapters would have to be tested. These could be inspired models proposed in existing literature or style or domain transfer.

This chapter also introduced optional weight sharing and investigates its detrimental effects on model performance. By not sharing elements of the Split Learning chain and thus not sharing part of the network with any other party, reconstruction attacks would theoretically become significantly more challenging although the practical increase of increase has not been investigated. Our results indicate that the network optimizes differently for the different weight sharing strategies, confirming that the links that are not shared vary from one another, but it comes at a cost in performance. This performance loss is significantly more prominent when the final link is not being shared. We hypothesize this to be due to the high complexity of the final fully connected layer and how it is closely connected to the final classification process.

Finally, we introduce the concept of vertical partitioned data for health care. To employ these opportunities would require a paradigm shift on what can be used as predictive medical data in the hospital. This comes with many technical hurdles that would have to be faced, with at the core the method used to fuse these data streams in a secure manner. We propose Split Learning as a viable option for such efforts. Although implementation, testing and designing real world use cases of such systems lies beyond the scope of this work, we propose Split Learning as a capable method for such challenges.

## 5.5 Conclusion

In addition to *vanilla* Split Learning, adaptations can be imagined to face specific problems. This chapter has discussed three.

Initially, we confirmed that inter-institutional heterogeneity can pose a challenge when training Split Learning models. We have proposed a novel Split Learning method that used local adapters to overcome this problem. Although it has shown promising results on initial tests further research is required to comprehensively investigate the potency of this method.

Secondly, we presented several alternative weight sharing strategies that should make reconstruction attacks more difficult. Although training of these models remained stable, performance dropped markedly.

Lastly, we have proposed Split Learning as a viable method to be used for managing prediction from multiple data streams, also known as *vertical partitioning*.

# 6 Conclusions

Privacy preserving distributed deep learning can enable health care institutions to collaborate on training deep learning models, improving their performance and generalizability.

Split Learning allows secure distributed training and inference of deep neural networks by sending features of data instead of raw data. By centralizing computational effort, it can also reduce computational power required for collaboration. Split Learning is functionally identical to training in centralized fashion and we have not observed dropping performance with an increasing number of participating clients. Communicational requirements of Split Learning are low enough to ensure feasibility in first world hospitals, even without overhead.

Challenges that Split Learning faces mainly center around training on heterogeneous data. Our work has shown heterogeneity to be detrimental to performance. However, our work has also proposed a novel method that employs local domain adapters to combat this effect. Although this method shows promising initial results, it requires further research into its feasibility.

Results on improving security using alternative weights sharing strategies mainly emphasize the cost in performance. Lastly, we advocate for Split Learning as a tool to integrate data flows from multiple sources in future research.

# Bibliography


1. Jiang F, Jiang Y, Zhi H, et al. Artificial intelligence in healthcare: past, present and future. *Stroke Vasc Neurol*. 2017;2(4):230-243.
2. Deng J, Dong W, Socher R, Li L, Kai Li, Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. ; 2009:248-255. doi:10.1109/CVPR.2009.5206848
3. Hinton G, Deng L, Yu D, et al. Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups. *IEEE Signal Process Mag*. 2012;29(6):82-97. doi:10.1109/MSP.2012.2205597
4. Collobert R, Weston J. A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. In: *Proceedings of the 25th International Conference on Machine Learning*. ICML '08. New York, NY, USA: ACM; 2008:160-167. doi:10.1145/1390156.1390177
5. Miotto R, Wang F, Wang S, Jiang X, Dudley JT. Deep learning for healthcare: review, opportunities and challenges. *Brief Bioinform*. 2018;19(6):1236-1246. doi:10.1093/bib/bbx044
6. Pooch EHP, Ballester PL, Barros RC. Can we trust deep learning models diagnosis? The impact of domain shift in chest radiograph classification. 2019.
7. Dluhoš P, Schwarz D, Cahn W, et al. Multi-center machine learning in imaging psychiatry: a meta-model approach. *Neuroimage*. 2017;155:10-24.
8. Ciresan DC, Meier U, Gambardella LM, Schmidhuber J. Deep Big Simple Neural Nets Excel on Handwritten Digit Recognition. *CoRR*. 2010;abs/1003.0. http://arxiv.org/abs/1003.0358.
9. Le Q V, Ngiam J, Coates A, Lahiri A, Prochnow B, Ng AY. On Optimization Methods for Deep Learning. In: *Proceedings of the 28th International Conference on International Conference on Machine Learning*. ICML'11. USA: Omnipress; 2011:265-272. http://dl.acm.org/citation.cfm?id=3104482.3104516.
10. Coates A, Ng A, Lee H. An Analysis of Single-Layer Networks in Unsupervised Feature Learning. In: Gordon G, Dunson D, Dudík M, eds. *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. Vol 15. Proceedings of Machine Learning Research. Fort Lauderdale, FL, USA: PMLR; 2011:215-223. http://proceedings.mlr.press/v15/coates11a.html.
11. Huang Y, Cheng Y, Bapna A, et al. GPipe: Efficient Training of Giant Neural Networks using Pipeline Parallelism. 2018.
12. Annas GJ. HIPAA regulations - A new era of medical-record privacy? *N Engl J Med*. 2003;348(15):1486-1490. doi:10.1056/NEJMlim035027
13. Mercuri RT. The HIPAA-potamus in Health Care Data Security. *Commun ACM*. 2004;47(7):25-28. doi:10.1145/1005817.1005840
14. Nass SJ, Levit LA, Gostin LO. *Beyond the HIPAA Privacy Rule: Enhancing Privacy, Improving Health Through Research*. (Nass SJ, Levit LA, Gostin LO, eds.). Washington (DC); 2009. doi:10.17226/12458
15. Luxton DD, Kayl RA, Mishkind MC. mHealth data security: the need for HIPAA-compliant standardization. *Telemed J E Health*. 2012;18(4):284-288. doi:10.1089/tmj.2011.0180
16. Porter CC. De-identified data and third party data mining: the risk of re-identification of personal information. *Shidler JL Com Tech*. 2008;5:1.
17. Sweeney L. k-anonymity: a model for protecting privacy. *Int J Uncertainty, Fuzziness Knowledge-Based Syst*. 2002;10(05):557-570. doi:10.1142/S0218488502001648

18. Sweeney L, Crosas M, Bar-Sinai M. Sharing sensitive data with confidence: The datatags system. *Technol Sci*. 2015.
19. Xia W, Wan Z, Yin Z, et al. It's all in the timing: calibrating temporal penalties for biomedical data sharing. *J Am Med Inform Assoc*. 2018;25(1):25-31. doi:10.1093/jamia/ocx101
20. Kim DW, Jang HY, Kim KW, Shin Y, Park SH. Design Characteristics of Studies Reporting the Performance of Artificial Intelligence Algorithms for Diagnostic Analysis of Medical Images: Results from Recently Published Papers. *Korean J Radiol*. 2019;20(3):405-410. https://doi.org/10.3348/kjr.2019.0025.
21. Hard A, Rao K, Mathews R, et al. Federated Learning for Mobile Keyboard Prediction. *CoRR*. 2018;abs/1811.0. http://arxiv.org/abs/1811.03604.
22. McMahan HB, Ramage D. Federated Learning: Collaborative Machine Learning without Centralized Training Data. https://ai.googleblog.com/2017/04/federated-learning-collaborative.html. Published 2017. Accessed October 12, 2019.
23. Gupta O, Raskar R. Distributed learning of deep neural network over multiple agents. *J Netw Comput Appl*. 2018;116:1-8. doi:https://doi.org/10.1016/j.jnca.2018.05.003
24. McMahan HB, Moore E, Ramage D, y Arcas BA. Communication-Efficient Learning of Deep Networks using Model Averaging. *CoRR*. 2016;abs/1602.0. http://arxiv.org/abs/1602.05629.
25. Evans J, Sadana R, Hillenback J, Licking E, Khanna S. *Pulse of the Industry: Medical Technology Report 2019*. London (UK); 2019. https://assets.ey.com/content/dam/ey-sites/ey-com/en_gl/topics/life-sciences/ey-pulse-of-the-industry-2019.pdf.
26. Jordan MI, Mitchell TM. Machine learning: Trends, perspectives, and prospects. *Science (80- )*. 2015;349(6245):255-260.
27. LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature*. 2015;521(7553):436-444. doi:10.1038/nature14539
28. Nations U. *World Population Ageing, 2013*. United Nations Publications; 2014.
29. van der Meulen M. Artificial Intelligence as a Driver of Value in Value-Based Health Care Systems. 2019.
30. Strong K, Mathers C, Leeder S, Beaglehole R. Preventing chronic diseases: how many lives can we save? *Lancet*. 2005;366(9496):1578-1582.
31. Association AH, others. When I'm 64: How boomers will change health care. *Chicago, Ill Am Hosp Assoc*. 2007:0-10.
32. Mwachofi A, Al-Assaf AF. Health care market deviations from the ideal market. *Sultan Qaboos Univ Med J*. 2011;11(3):328.
33. Abbo ED, Zhang Q, Zelder M, Huang ES. The increasing number of clinical items addressed during the time of adult primary care visits. *J Gen Intern Med*. 2008;23(12):2058.
34. Jha S, Topol EJ. Adapting to Artificial Intelligence: Radiologists and Pathologists as Information Specialists. *JAMA*. 2016;316(22):2353-2354. doi:10.1001/jama.2016.17438
35. Patel VL, Shortliffe EH, Stefanelli M, et al. The coming of age of artificial intelligence in medicine. *Artif Intell Med*. 2009;46(1):5-17. doi:10.1016/j.artmed.2008.07.017
36. Jha S, Topol EJ. Adapting to Artificial Intelligence: Radiologists and Pathologists as Information Specialists. *JAMA*. 2016;316(22):2353-2354. doi:10.1001/jama.2016.17438
37. Bhargavan M, Sunshine JH. Utilization of radiology services in the United States: levels and trends in modalities, regions, and populations. *Radiology*. 2005;234(3):824-832. doi:10.1148/radiol.2343031536
38. Lu Y, Zhao S, Chu PW, Arenson RL. An update survey of academic radiologists' clinical productivity. *J Am Coll Radiol*. 2008;5(7):817-826. doi:10.1016/j.jacr.2008.02.018
39. Berlin L. Liability of interpreting too many radiographs. *AJR Am J Roentgenol*. 2000;175(1):17-22. doi:10.2214/ajr.175.1.1750017
40. Fitzgerald R. Error in radiology. *Clin Radiol*. 2001;56(12):938-946.
41. Nakajima Y, Yamada K, Imamura K, Kobayashi K. Radiologist supply and workload: international comparison. *Radiat Med*. 2008;26(8):455-465.
42. Zheng Y, He M, Congdon N. The worldwide epidemic of diabetic retinopathy. *Indian J Ophthalmol*. 2012;60(5):428.
43. Litjens G, Kooi T, Bejnordi BE, et al. A survey on deep learning in medical image analysis. *Med Image Anal*.

2017;42:60-88.
44. Wong TY, Bressler NM. Artificial intelligence with deep learning technology looks into diabetic retinopathy screening. *Jama*. 2016;316(22):2366-2367.
45. Gillies RJ, Kinahan PE, Hricak H. Radiomics: Images Are More than Pictures, They Are Data. *Radiology*. 2016;278(2):563-577. doi:10.1148/radiol.2015151169
46. Fogel AL, Kvedar JC. Artificial intelligence powers digital medicine. *npj Digit Med*. 2018;1(1):5. doi:10.1038/s41746-017-0012-2
47. Topol E. *Deep Medicine: How Artificial Intelligence Can Make Healthcare Human Again*. Hachette UK; 2019.
48. Singh H, Graber ML. Improving Diagnosis in Health Care--The Next Imperative for Patient Safety. *N Engl J Med*. 2015;373(26):2493.
49. Bruno MA, Walker EA, Abujudeh HH. Understanding and confronting our mistakes: the epidemiology of error in radiology and strategies for error reduction. *Radiographics*. 2015;35(6):1668-1676.
50. Donaldson MS, Corrigan JM, Kohn LT, others. *To Err Is Human: Building a Safer Health System*. Vol 6. National Academies Press; 2000.
51. Lasic M. *Case Study: An Isulin Overdose*. Boston (MA); 2018. http://www.ghbook.ir/index.php?name=فرهنگ و رسانه های نوین&option=com_dbook&task=readonline&book_id=13650&page=73&chkhashk=ED9C9491B4&Itemid=218&lang=fa&tmpl=component.
52. Dilsizian SE, Siegel EL. Artificial intelligence in medicine and cardiac imaging: harnessing big data and advanced computing to provide personalized medical diagnosis and treatment. *Curr Cardiol Rep*. 2014;16(1):441. doi:10.1007/s11886-013-0441-8
53. Murdoch TB, Detsky AS. The inevitable application of big data to health care. *JAMA*. 2013;309(13):1351-1352. doi:10.1001/jama.2013.393
54. Kolker E, Ozdemir V, Kolker E. How Healthcare Can Refocus on Its Super-Customers (Patients, n =1) and Customers (Doctors and Nurses) by Leveraging Lessons from Amazon, Uber, and Watson. *OMICS*. 2016;20(6):329-333. doi:10.1089/omi.2016.0077
55. Syed Z, Stultz CM, Scirica BM, Guttag J V. Computationally generated cardiac biomarkers for risk stratification after acute coronary syndrome. *Sci Transl Med*. 2011;3(102):102ra95. doi:10.1126/scitranslmed.3002557
56. Wang J, Zhang L, Liu T, Pan W, Hu B, Zhu T. Acoustic differences between healthy and depressed people: a cross-situation study. *BMC Psychiatry*. 2019;19(1):300. doi:10.1186/s12888-019-2300-7
57. Taguchi T, Tachikawa H, Nemoto K, et al. Major depressive disorder discrimination using vocal acoustic features. *J Affect Disord*. 2018;225:214-220.
58. Simpson JT, Us TN. SENTIMENT ANALYSIS OF MENTAL HEALTH DISORDER SYMPTOMS. 2019;2(12).
59. Dam NT Van, O'Connor D, Marcelle ET, et al. Data-Driven Phenotypic Categorization for Neurobiological Analyses: Beyond DSM-5 Labels. *Biol Psychiatry*. 2017;81(6):484-494. doi:https://doi.org/10.1016/j.biopsych.2016.06.027
60. Li L, Cheng W-Y, Glicksberg BS, et al. Identification of type 2 diabetes subgroups through topological analysis of patient similarity. *Sci Transl Med*. 2015;7(311):311ra174--311ra174. doi:10.1126/scitranslmed.aaa9364
61. McCorduck P, Cfe C. *Machines Who Think: A Personal Inquiry into the History and Prospects of Artificial Intelligence*. CRC Press; 2004.
62. Goodfellow I, Bengio Y, Courville A. *Deep Learning*. MIT press; 2016.
63. Russakovsky O, Deng J, Su H, et al. Imagenet large scale visual recognition challenge. *Int J Comput Vis*. 2015;115(3):211-252.
64. Lee J-G, Jun S, Cho Y-W, et al. Deep learning in medical imaging: general overview. *Korean J Radiol*. 2017;18(4):570-584.
65. Szegedy C, Wei Liu, Yangqing Jia, et al. Going deeper with convolutions. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. ; 2015:1-9. doi:10.1109/CVPR.2015.7298594
66. Matsugu M, Mori K, Mitari Y, Kaneda Y. Subject independent facial expression recognition with robust face

detection using a convolutional neural network. *Neural Networks*. 2003;16(5-6):555-559.
67. Fukushima K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol Cybern*. 1980;36(4):193-202.
68. Russell SJ, Norvig P. *Artificial Intelligence: A Modern Approach*. Malaysia; Pearson Education Limited,; 2016.
69. Nilsson NJ. *Principles of Artificial Intelligence*. Morgan Kaufmann; 2014.
70. Erenshteyn R, Foulds R, Galuska S. Is Designing a Neural Network Application an Art or a Science? *SIGCHI Bull*. 1994;26(3):23-29. doi:10.1145/181518.181522
71. Snoek J, Rippel O, Swersky K, et al. Scalable bayesian optimization using deep neural networks. In: *International Conference on Machine Learning*. ; 2015:2171-2180.
72. Choromanska A, Henaff M, Mathieu M, Arous G Ben, LeCun Y. The loss surfaces of multilayer networks. In: *Artificial Intelligence and Statistics*. ; 2015:192-204.
73. Maglogiannis IG. *Emerging Artificial Intelligence Applications in Computer Engineering: Real Word Ai Systems with Applications in Ehealth, Hci, Information Retrieval and Pervasive Technologies*. Vol 160. Ios Press; 2007.
74. Schmidt-Erfurth U, Sadeghipour A, Gerendas BS, Waldstein SM, Bogunović H. Artificial intelligence in retina. *Prog Retin Eye Res*. 2018;67:1-29. doi:https://doi.org/10.1016/j.preteyeres.2018.07.004
75. Gibson E, Giganti F, Hu Y, et al. Automatic multi-organ segmentation on abdominal CT with dense v-networks. *IEEE Trans Med Imaging*. 2018;37(8):1822-1834.
76. Larson DB, Chen MC, Lungren MP, Halabi SS, Stence N V, Langlotz CP. Performance of a deep-learning neural network model in assessing skeletal maturity on pediatric hand radiographs. *Radiology*. 2017;287(1):313-322.
77. Ross E, Shah N, Leeper N. Phenotype Discovery in Cardiovascular Patients Using Unsupervised Learning. *Arterioscler Thromb Vasc Biol*. 2018;38(Suppl_1):A245-A245. doi:10.1161/atvb.38.suppl_1.245
78. Cui J, Gong K, Guo N, et al. PET image denoising using unsupervised deep learning. *Eur J Nucl Med Mol Imaging*. August 2019. doi:10.1007/s00259-019-04468-4
79. Tomašev N, Glorot X, Rae JW, et al. A clinically applicable approach to continuous prediction of future acute kidney injury. *Nature*. 2019;572(7767):116-119. doi:10.1038/s41586-019-1390-1
80. Gianfrancesco MA, Tamang S, Yazdany J, Schmajuk G. Potential Biases in Machine Learning Algorithms Using Electronic Health Record Data. *JAMA Intern Med*. 2018;178(11):1544-1547. doi:10.1001/jamainternmed.2018.3763
81. Sheller MJ, Reina GA, Edwards B, Martin J, Bakas S. Multi-institutional Deep Learning Modeling Without Sharing Patient Data: A Feasibility Study on Brain Tumor Segmentation. In: Crimi A, Bakas S, Kuijf H, Keyvan F, Reyes M, van Walsum T, eds. *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*. Cham: Springer International Publishing; 2019:92-104.
82. Dluhos P, Schwarz D, Cahn W, et al. Multi-center machine learning in imaging psychiatry: A meta-model approach. *Neuroimage*. 2017;155:10-24. doi:10.1016/j.neuroimage.2017.03.027
83. IBM Watson Health. Understanding AI's fundamental value to healthcare. 2018.
84. Derrington D. Artificial intelligence for health and health care. *See https//www Heal gov/sites/default/files/jsr-17-task-002_aiforhealthandhealthcare12122017 pdf (last checked 8 Novemb 2018)*. 2017.
85. Leung MKK, Delong A, Alipanahi B, Frey BJ. Machine Learning in Genomic Medicine: A Review of Computational Problems and Data Sets. *Proc IEEE*. 2016;104(1):176-197. doi:10.1109/JPROC.2015.2494198
86. Nuffield Council on Bioethics. Bioethics Briefing Note: AI in Healthcare and Research. 2018:1-8. http://nuffieldbioethics.org/wp-content/uploads/Artificial-Intelligence-AI-in-healthcare-and-research.pdf.
87. Philips. Using AI to meet operational , clinical goals. *Philips Exec Insights*. 2018;(February).
88. Chervenak A, Foster I, Kesselman C, Salisbury C, Tuecke S. The data grid: Towards an architecture for the distributed management and analysis of large scientific datasets. *J Netw Comput Appl*. 2000;23(3):187-200.
89. Dean J, Corrado G, Monga R, et al. Large Scale Distributed Deep Networks. In: Pereira F, Burges CJC, Bottou L, Weinberger KQ, eds. *Advances in Neural Information Processing Systems 25*. Curran Associates, Inc.; 2012:1223-1231. http://papers.nips.cc/paper/4687-large-scale-distributed-deep-networks.pdf.
90. Yu KH, Beam AL, Kohane IS. Artificial intelligence in healthcare. *Nat Biomed Eng*. 2018;2(10):719-731.

doi:10.1038/s41551-018-0305-z
91. Sox HC. *Assessment of Diagnostic Technology in Health Care: Rationale, Methods, Problems, and Directions*. National Academies; 1989.
92. Meinert CL. Toward more definitive clinical trials. *Control Clin Trials*. 1980;1(3):249-261. doi:10.1016/0197-2456(80)90005-7
93. O'Sullivan PS, Stoddard HA, Kalishman S. Collaborative research in medical education: a discussion of theory and practice. *Med Educ*. 2010;44(12):1175-1184.
94. STAGE ION. Guidelines for Multi-Institutional/Collaborative Research.
95. Chonka A, Xiang Y, Zhou W, Bonti A. Cloud security defence to protect cloud computing against HTTP-DoS and XML-DoS attacks. *J Netw Comput Appl*. 2011;34(4):1097-1107.
96. Wu B, Wu J, Fernandez EB, Magliveras S. Secure and efficient key management in mobile ad hoc networks. In: *19th IEEE International Parallel and Distributed Processing Symposium*. ; 2005:8--pp.
97. Secretan J, Georgiopoulos M, Castro J. A privacy preserving probabilistic neural network for horizontally partitioned databases. In: *2007 International Joint Conference on Neural Networks*. ; 2007:1554-1559.
98. Zinkevich M, Weimer M, Li L, Smola AJ. Parallelized Stochastic Gradient Descent. In: Lafferty JD, Williams CKI, Shawe-Taylor J, Zemel RS, Culotta A, eds. *Advances in Neural Information Processing Systems 23*. Curran Associates, Inc.; 2010:2595-2603. http://papers.nips.cc/paper/4006-parallelized-stochastic-gradient-descent.pdf.
99. Mcdonald R, Mohri M, Silberman N, Walker D, Mann GS. Efficient Large-Scale Distributed Training of Conditional Maximum Entropy Models. In: Bengio Y, Schuurmans D, Lafferty JD, Williams CKI, Culotta A, eds. *Advances in Neural Information Processing Systems 22*. Curran Associates, Inc.; 2009:1231-1239. http://papers.nips.cc/paper/3881-efficient-large-scale-distributed-training-of-conditional-maximum-entropy-models.pdf.
100. Agarwal A, Duchi JC. Distributed delayed stochastic optimization. In: *Advances in Neural Information Processing Systems*. ; 2011:873-881.
101. Agarwal A, Chapelle O, Dud\'\ik M, Langford J. A reliable effective terascale linear learning system. *J Mach Learn Res*. 2014;15(1):1111-1133.
102. Barni M, Orlandi C, Piva A. A privacy-preserving protocol for neural-network-based computation. In: *Proceedings of the 8th Workshop on Multimedia and Security*. ; 2006:146-151.
103. Pathak D, Krahenbuhl P, Donahue J, Darrell T, Efros AA. Context encoders: Feature learning by inpainting. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. ; 2016:2536-2544.
104. Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol P-A. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J Mach Learn Res*. 2010;11(Dec):3371-3408.
105. Phong LT, Aono Y, Hayashi T, Wang L, Moriai S. Privacy-preserving deep learning via additively homomorphic encryption. *IEEE Trans Inf Forensics Secur*. 2018;13(5):1333-1345.
106. Wu D, Haven J. Using homomorphic encryption for large scale statistical analysis. 2012.
107. Lu W, Kawasaki S, Sakuma J. Using Fully Homomorphic Encryption for Statistical Analysis of Categorical, Ordinal and Numerical Data. *IACR Cryptol ePrint Arch*. 2016;2016:1163.
108. Orlandi C, Piva A, Barni M. Oblivious neural network computing via homomorphic encryption. *EURASIP J Inf Secur*. 2007;2007(1):37343.
109. Liu J, Juuti M, Lu Y, Asokan N. Oblivious neural network predictions via minionn transformations. In: *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ; 2017:619-631.
110. Vepakomma P, Swedish T, Raskar R, Gupta O, Dubey A. No Peek: {A} Survey of private distributed deep learning. *CoRR*. 2018;abs/1812.0. http://arxiv.org/abs/1812.03288.
111. French RM. Catastrophic forgetting in connectionist networks. *Trends Cogn Sci*. 1999;3(4):128-135.
112. Kirkpatrick J, Pascanu R, Rabinowitz N, et al. Overcoming catastrophic forgetting in neural networks. *Proc Natl Acad Sci*. 2017;114(13):3521-3526.
113. Chang K, Balachandar N, Lam C, et al. Distributed deep learning networks among institutions for medical imaging. *J Am Med Inform Assoc*. 2018;25(8):945-954. doi:10.1093/jamia/ocy017

114. Vepakomma P, Gupta O, Swedish T, Raskar R. Split learning for health: Distributed deep learning without sharing raw patient data. *CoRR*. 2018;abs/1812.0. http://arxiv.org/abs/1812.00564.
115. Vepakomma P, Gupta O, Dubey A, Raskar R. Reducing leakage in distributed deep learning for sensitive health data. *arXiv Prepr arXiv181200564*. 2019.
116. Hsieh K, Phanishayee A, Mutlu O, Gibbons PB. The Non-IID Data Quagmire of Decentralized Machine Learning. 2019.
117. Poirot M. SplitLearning GitHub Repository. https://github.com/MGPoirot/SplitLearning. Published 2019.
118. Tapp RJ, Shaw JE, Harper CA, et al. The Prevalence of and Factors Associated With Diabetic Retinopathy in the Australian Population. *Diabetes Care*. 2003;26(6):1731-1737. doi:10.2337/diacare.26.6.1731
119. Diabetic Retinopathy Detection. https://www.kaggle.com/c/diabetic-retinopathy-detection. Published 2015. Accessed November 9, 2019.
120. He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition. *2016 IEEE Conf Comput Vis Pattern Recognit*. 2016:770-778.
121. Fyllingen EH, Stensjøen AL, Berntsen EM, Solheim O, Reinertsen I. Glioblastoma Segmentation: Comparison of Three Different Software Packages. *PLoS One*. 2016;11(10):1-16. doi:10.1371/journal.pone.0164891
122. Menze BH, Jakab A, Bauer S, et al. The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS). *IEEE Trans Med Imaging*. 2015;34(10):1993-2024. doi:10.1109/TMI.2014.2377694
123. Bakas S, Akbari H, Sotiras A, et al. Advancing The Cancer Genome Atlas glioma MRI collections with expert segmentation labels and radiomic features. *Sci data*. 2017;4:170117. doi:10.1038/sdata.2017.117
124. Bakas S, Reyes M, Jakab A, et al. Identifying the Best Machine Learning Algorithms for Brain Tumor Segmentation, Progression Assessment, and Overall Survival Prediction in the BRATS Challenge. 2018.
125. Ronneberger O, Fischer P, Brox T. U-Net: Convolutional Networks for Biomedical Image Segmentation. 2015.
126. Irvin J, Rajpurkar P, Ko M, et al. CheXpert: A Large Chest Radiograph Dataset with Uncertainty Labels and Expert Comparison. 2019. http://arxiv.org/abs/1901.07031.
127. Huang G, Liu Z, Weinberger KQ. Densely Connected Convolutional Networks. *CoRR*. 2016;abs/1608.0. http://arxiv.org/abs/1608.06993.
128. Rajpurkar P, Irvin J, Zhu K, et al. CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning. *CoRR*. 2017;abs/1711.0. http://arxiv.org/abs/1711.05225.
129. Rajpurkar P, Irvin J, Bagul A, et al. MURA: Large Dataset for Abnormality Detection in Musculoskeletal Radiographs. 2017.
130. Pho K. SplitLearning_Private GitHub Reposity. https://github.com/OmnInfinity/SplitLearning_Private. Published 2019.
131. Singh A, Vepakomma P, Gupta O, Raskar R. Detailed comparison of communication efficiency of split learning and federated learning. 2019.
132. Strawn B. How Fast is Fiber? https://www.highspeedinternet.com/resources/how-fast-is-fiber. Published 2015. Accessed November 25, 2019.
133. Darrell T, Kloft M, Pontil M, Rätsch G, Rodner E. Machine Learning with Interdependent and Non-identically Distributed Data (Dagstuhl Seminar 15152). In: *Dagstuhl Reports*. Vol 5. ; 2015.
134. Hazan A, Shoshan Y, Khapun D, Aladjem R, Ratner V. AdapterNet - learning input transformation for domain adaptation. 2018.
135. Graham B. *Diabetic Retinopathy Detection Competition Report.*; 2015. https://www.kaggle.com/c/diabetic-retinopathy-detection/discussion/15801.
136. Boonstra TW, Nicholas J, Wong QJJ, Shaw F, Townsend S, Christensen H. Using mobile phone sensor technology for mental health research: Integrated analysis to identify hidden challenges and potential solutions. *J Med Internet Res*. 2018;20(7):e10131.
137. Rosenfeld A, Benrimoh D, Armstrong C, et al. Big Data Analytics and AI in Mental Healthcare. *arXiv Prepr arXiv190312071*. 2019.
138. Spring B, Gotsis M, Paiva A, Spruijt-Metz D. Healthy apps: mobile devices for continuous monitoring and intervention. *IEEE Pulse*. 2013;4(6):34-40.
139. Rauber J, Fox EB, Gatys LA. Modeling patterns of smartphone usage and their relationship to cognitive health.

2019.
140. Lee D, Kim J, Moon W-J, Ye JC. CollaGAN: Collaborative GAN for missing image data imputation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. ; 2019:2487-2496.
141. Hagiwara A, Otsuka Y, Hori M, et al. Improving the Quality of Synthetic FLAIR Images with Deep Learning Using a Conditional Generative Adversarial Network for Pixel-by-Pixel Image Translation. *Am J Neuroradiol*. 2019. doi:10.3174/ajnr.A5927
142. RSNA. RSNA Intracranial Hemorrhage Detection. https://www.kaggle.com/c/rsna-intracranial-hemorrhage-detection. Published 2019.
143. Kingma DP, Ba J. Adam: A method for stochastic optimization. *arXiv Prepr arXiv14126980*. 2014.
144. Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks. In: *In Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS'10). Society for Artificial Intelligence and Statistics*. ; 2010.
145. Agrahari R. DenseNet on MURA Dataset using PyTorch. https://github.com/pyaf/DenseNet-MURA-PyTorch. Published 2018. Accessed January 1, 2020.

# A Appendix

## 8.1 Data set and implementation details

**Brain Tumor Segmentation (BraTS):** The BraTS data set poses a segmentation problem of gliomata from magnetic resonance imaging (MRI) brain scans. The data set consists of 259 pre-processed T2 Fluid Attenuated Inversion Recovery (FLAIR) scans acquired according to different clinical protocols, from various scanners and from multiple sites, from cases presenting high grade glioma (HGG). Segmentation labels were created by manually by one to four raters and were approved by experienced neuro-radiologists. Original segmentation labels consisted of the gadolinium-enhancing tumor, peritumoral edema and necrotic and non-enhancing tumor core. We only considered the whole tumor volume as defined by the union of the three labels to allow for simplified evaluation of the method. Data sets were partitioned into 75% training, 15% validation and 10% independent test data sets without any patients overlapping partitions resulting in 194 training, 39 validation and 26 test samples. Training and validation partitions were equally distributed over the number of participating institutions. Test performance was computed on the entire test partition using the model state with lowest loss on the validation set.

A voxel wise binary cross entropy (BCE) was used. Adam optimization[143] using standard parameters ($\beta_1 = 0.9$ and $\beta_2 = 0.999$), and learning rate of $5^{-4}$ without decay was used. Batch size used was 16. Data was augmented by 50% chance of lateral inversion. The system was implemented in Python (version 3.7.5) and PyTorch (version 1.2.0). Training was performed on a GeForce GTX TITAN X graphics processing unit until validation accuracy reached a plateau as defined by not decreasing for more than 30 epochs.

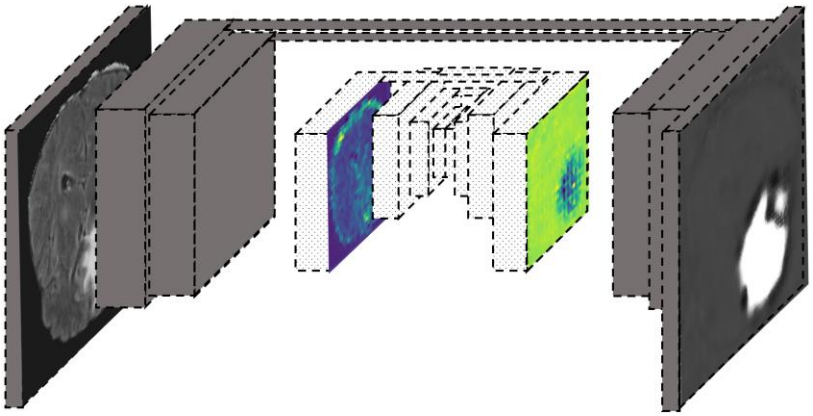


*Figure 17: Schematic of proposed Split Learning adaptation of a U-Net*

**Chest X-ray multi label classification (CheXpert)**: The CheXpert data presents a multi-label classification problem of fourteen common chest radiographic observations from a large set of chest radiographs. The data set consists of 224,316 chest radiographs with labels of 65,240 patients. Labels were generated using natural language processing (NLP). Cases where labels contained uncertainty were excluded according to the baseline approach as described in the paper[126]. All frontal images of most commonly occurring (320x390 px) resolution were used leaving a remaining dataset of 96,326 chest radiographs. Data sets were partitioned into 75% training, 15% validation and 10% independent test data sets without any patients overlapping partitions resulting in 72,244 training, 14,449 validation and 9,633 test samples. Training and validation partitions were equally distributed over the number of participating institutions. Test performance was computed on the entire test partition using the model state with lowest loss on the validation set.

The network was pretrained on ImageNet[2], loss was defined computed using a combined sigmoid BCE loss. Adam optimization[143] using standard parameters, and default learning rate $10^{-4}$ without decay was used. Batch size used was 32. Data was augmented by 50% chance of lateral inversion. The system was implemented in Python (version 3.7.5) and PyTorch (version 1.2.0). Training was performed on a Nvidia GeForce GTX 1080 Ti graphics processing unit. Validation was performed after 10% of every epoch. Models were trained until validation loss had not decreased after 30 validations. The state of lowest validation loss was used for inference on the test set.

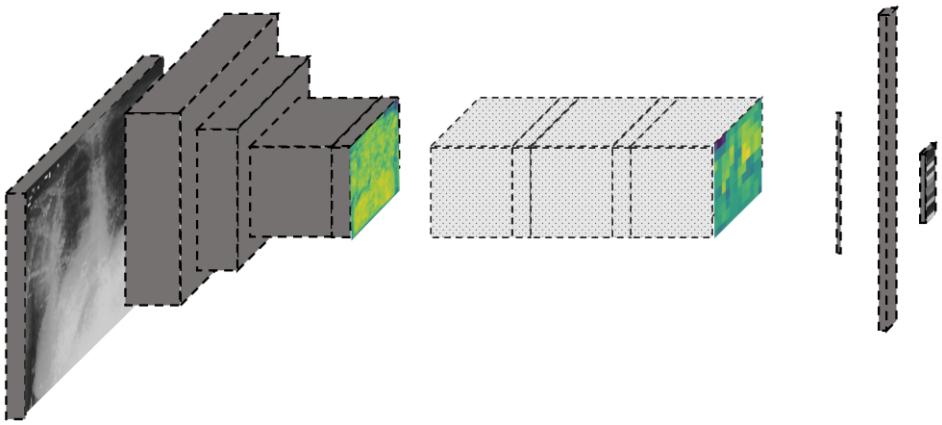


*Figure 18: Schematic of proposed Split Learning adaptation of a DenseNet*

**Diabetic retinopathy challenge (DRC) binary classification:** We used the diabetic retinopathy challenge (DRC) dataset as previously. This data set originates from the Kaggle Diabetic Retinopathy dataset of retinal fundus photos. The original multi-class classification problem was simplified to binary classification y ∈ {0, 1} indicating normal or abnormal respectively. A class-balanced subset of 9000 images was used for training and validation to prevent saturation of learning for models. Images were down sampled to 256x256 RGB images. The images were pre-processed via the method detailed in the competition report by the winner.[135] This included high pass filtering to account for image capturing variation, cropping to remove filtering artifacts and histogram normalization.

A 34-layer residual network[120] (Resnet-34) architecture was utilized with Glorot uniform initialization[144], stochastic gradient descent (SGD) optimization using standard parameters, and default learning rate $10^4$ without decay was used. Data was augmented in real-time using random rotations (0-360°) and 50% chance of lateral or axial inversion. Loss was computed used a binary cross entropy loss function. Training was performed on a GeForce GTX TITAN X graphics processing unit until validation accuracy reached a plateau as defined by not decreasing for more than 30 epochs.

**Musculoskeletal Radiograph binary classification:** MURA is a large data set containing 40,009 bone X-rays images from 14,052 studies. Each image was labeled as either normal or abnormal by radiologists, presenting a binary classification problem. From the total number of studies, a random subset was retained to include both 5177 positive as negative cases. Instead of the DenseNet commonly used in literature for this problem, a 152-layer residual network was implemented, and split much like the network for the DRC data set. A residual network was used to also include models with a comparatively higher number of parameters, as to diversify the four tasks implemented. Data augmentation was based on similar methods found in literature[145], which consisted of random lateral inversion, 10° of random rotation, normalization and resizing to 224×22 pixels. Loss was computed as classification accuracy. Training was performed on a GeForce GTX TITAN X graphics processing unit until validation accuracy reached a plateau as defined by not decreasing for more than 20 epochs.

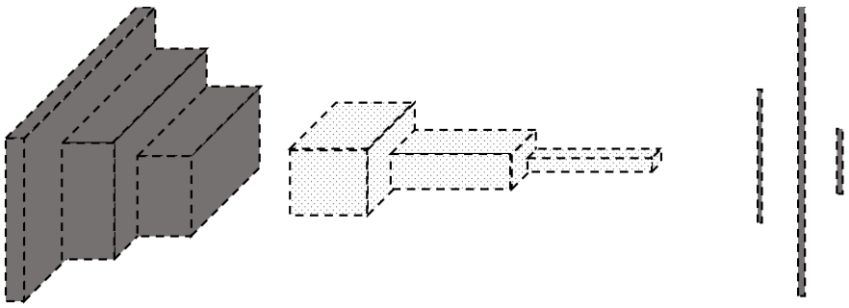


*Figure 19: Schematic of proposed Split Learning adaptation of ResNet*

## 8.2 Split Learning Algorithm

**Algorithm 1**: Boomerang Split Learning

Server Side:

| | | |
|---|---|---|
| 1: | $H \leftarrow \{h_A, h_B, \dots, h_Z\}$ | Assign participating hospitals. |
| 2: | $F \leftarrow \{L_0, L_1, \dots, L_N\}$ | Define neural network architecture. |
| 3: | $G \leftarrow objective\ function$ | Define the objective function |
| 4: | $F_{front}, F_{center}, F_{back} \leftarrow \{L_{0\rightarrow n}\}, \{L_{n+1\rightarrow m}\}, \{L_{m+1\rightarrow N}\}$ | Split network. |
| 5: | for $h$ in $H$ do | |
| 6: | $F^h_{front}, F^h_{back} \leftarrow F_{front}, F_{back}$ | Assign model states. |
| 7: | while $h$ contains more unique samples do | |
| 8: | $F^h \leftarrow$ TRAIN_NETWORK($h$) | Train neural network. |
| 9: | $F_{front}, F_{back} \leftarrow F^h_{front}, F^h_{back}$ | Update model states. |
| | | |
| 0: | procedure TRAIN_NETWORK($h$) | |
| 1: | $X_n \leftarrow h$.FORWARD_PASS( ) | Retrieve features of sample X. |
| 2: | $X_m \leftarrow F_{center}(X_n)$ | Propagate features up to $L_m$ |
| 3: | $F_{back}, \nabla_m \leftarrow h$.CENTER_PASS($X_m$) | Send $m^{th}$ layer features to hospital. |
| 4: | $F_{center}, \nabla_n \leftarrow F_{center}(\nabla_m)$ | Apply gradients up to $L_{n+1}$. |
| 5: | $F_{front} \leftarrow h$.BACK_PASS($\nabla_n$) | Send $n+1^{st}$ gradients to hospital |
| 6: | return $F^h$ | |

Institution Side:

| | | |
|---|---|---|
| 0: | procedure FORWARD_PASS | |
| 1: | $X_0, Y \leftarrow$ a unique sample-label pair | Get unique data sample |
| 2: | $X_n = F^h_{front}(X_0)$ | Propagate data up to $L_n$ |
| 3: | return $X_n$ | Send $n^{th}$ layer features to server |
| | | |
| 0: | procedure CENTER_PASS($X_m$) | |
| 1: | $\hat{Y} \leftarrow F^h_{back}(X_m)$ | Propagate features up to $L_N$ |
| 2: | $\nabla_N \leftarrow G(\hat{Y}, Y)$ | Compute gradients. |
| 3: | $F^h_{back}, \nabla_m = F^h_{back}(\nabla_N)$ | Apply gradients up to $L_{m+1}$. |
| 4: | return $F^h_{back}, \nabla_m$ | Send gradients to server. |
| | | |
| 0: | procedure BACK_PASS($\nabla_n$) | |
| 1: | $F^h_{front} = F^h_{front}(\nabla_n)$ | Apply gradients up to $L_0$. |
| 2: | return $F^h_{front}$ | |

## 8.3 Split Learning with Local Adapters Algorithm

**Algorithm 2**: Boomerang Split Learning with Local Adapters

Server Side:

$H \leftarrow \{h_A, h_B, \ldots, h_Z\}$

$F \leftarrow \{L_0, L_1, \ldots, L_N\}$

$G \leftarrow objective\ function$

$F_{front}, F_{center}, F_{back} \leftarrow \{L_{0 \to n}\}, \{L_{n+1 \to m}\}, \{L_{m+1 \to N}\}$

for $h$ in $H$ do

    $F^h_{front}, F^h_{back} \leftarrow F_{front}, F_{back}$

    while performance of $F^h$ increases do — As long as it improves performance

        $F^h_{front} \leftarrow$ TRAIN_NETWORK($h$) — Train the front node

    while $h$ contains more unique samples do

        $F^h \leftarrow$ TRAIN_NETWORK($h$)

    $F_{front}, F_{back} \leftarrow F^h_{front}, F^h_{back}$