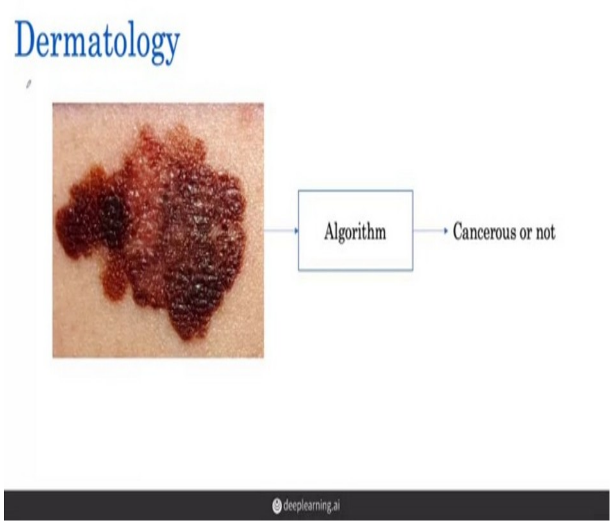
Artificial Intelligence For Medical Diagnosis

Diagnosis is about identifying disease , in these part we will see

- Building an algorithm to look to an X - ray and determine whether it can contain a disease
- Build an algorithm that looks Brain's MRI and identify the location of tumors in those Brain's MRI

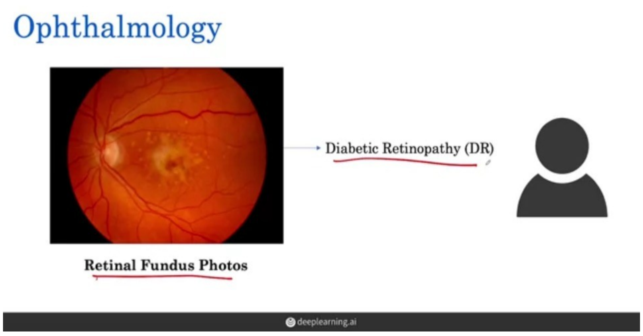
In the next part we will see about Medical Prognosis:- Which means the process of predicting the future health of the patients, this problem is mostly solved using structured data , having the lab results or the demographic of the patient and use that to predict on how much that patient is at risk of death or at risk of heart attack.

Dermatologist is a branch of medicine dealing with a skin , one task of dermatologist is to look at a suspicious region of a skin to determine whether a mole skin cancer or not.

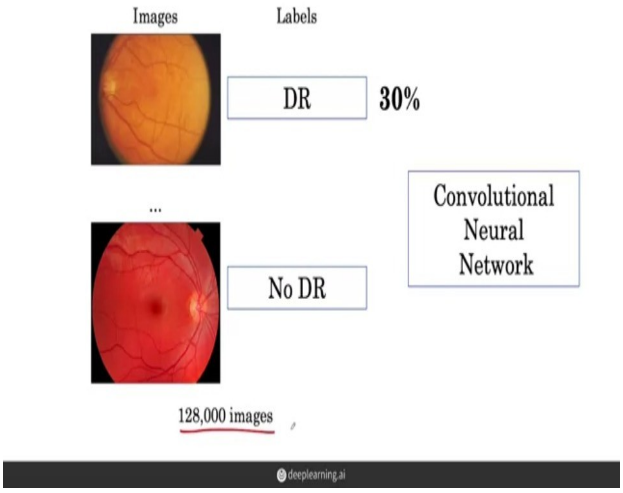


So an Algorithm is trained to determine whether a region of a skin tissue is cancerous or not , using hundreds of thousands of images and labels as input , a convolutional neural network can be trained to do this task. Once the algorithm has been trained, the algorithms predictions can be evaluated against the predictions of human dermatologists on a new set of images. In this study, it was found that the algorithm performed as well as the dermatologists did.

Our second example is ophthalmology which deals with the diagnosis and treatment of eye disorder. one disease or pathology to look at here is diabetic retinopathy which is damage to retina caused by diabetes and is a measure of diabetes. Currently diabetes retinopathy (DR) is a time consuming and manual process that requires a trained clinician to examine these photos. The algorithm was developed to determine whether patients had diabetic retinopathy by looking at such photos. What does it mean by Pathology ? It's a science which deals about causes and effects of disease , they are mostly people's who work in laboratory.

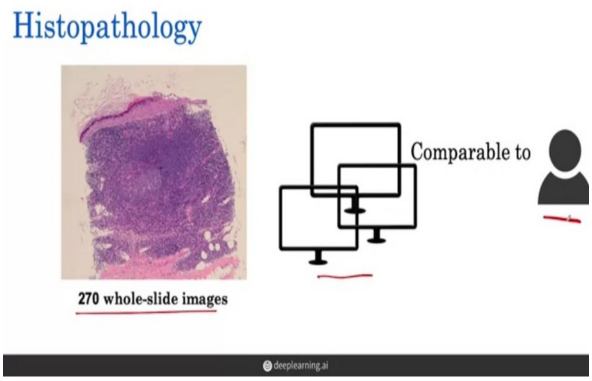


This study used over 128,000 images of which only 30 percent had diabetic retinopathy. we will look at this data imbalanced problem , which is prominent in medicine and so many other fields with real world data which means in real world most of the patients are negative to retinopathy, so our data will be more imbalanced.

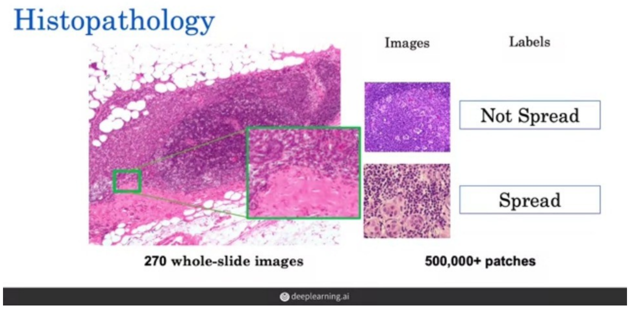


The study showed that the performance of the resulting algorithm was comparable to ophthalmologists. What does it mean human level performance ? as we know it depends may be it will be a young ophthalmologists , or a group of specialized ophthalmologists.

Our third example is histopathology a medical specialty involving examination of tissues under the microscope. One of the tasks that pathologist do is to look at the scanned microscopic images of tissue called whole slide images and determine the extent to which a cancer has spread.



In one study in 2017 using only 270 whole slide images AI algorithms were developed and then evaluated against pathologists. It was found that the best algorithms performed as well as the pathologists did.

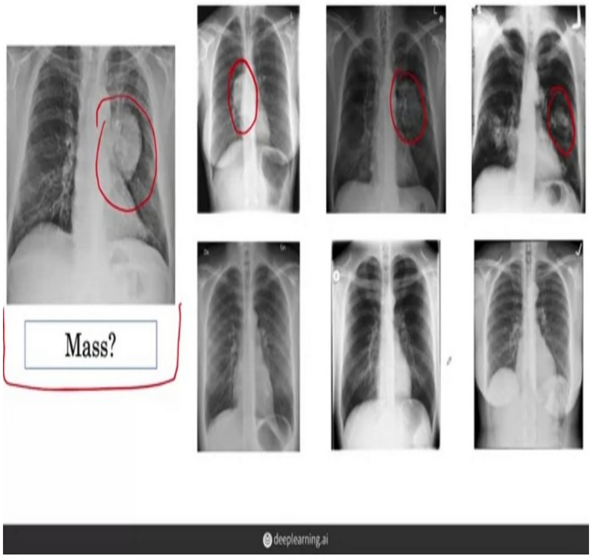


Now in histopathology , the images are very large and can not be fed directly in to an algorithm without breaking them down. WHY ? Because of computational complexity and over fitting , suppose we used a standard neural networks and we have one large 10,000 * 10,000 pixels input image , if the first layer has 300 neurons, which means only in the first layer we will have 30 million parameters only in the first layer, these leads to overfitting. The general step of these studies is that instead of feeding in one large , high resolution digital image of the slide , several patches are extracted at high magnification and used to train a model. These patches are labeled with the original label of the whole slide image and then fed in to a deep learning algorithm. In this way the algorithm can be trained on hundreds of thousands of patches. In this course , we will apply similar idea of breaking down a large image in to smaller images for model training to the task of brain tumor segmentation.

Building and Training For Medical Diagnosis

Now that you've seen some of the cutting-edge applications of deep learning to medical image classification problems, we'll look at how you can build your own deep learning model for the medical imaging task of using chest X-rays to detect multiple diseases with a single model.

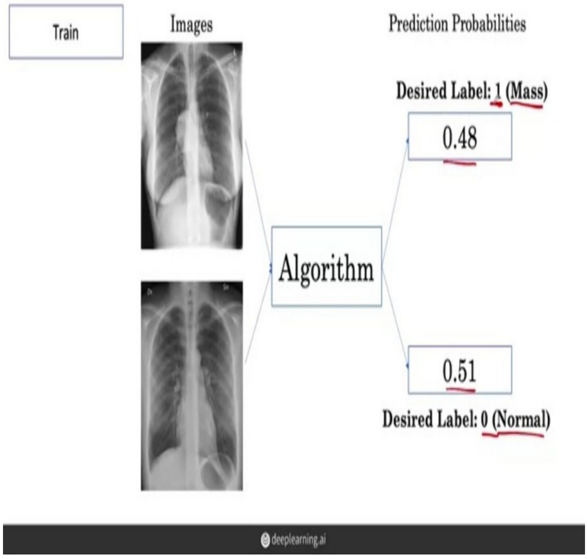
How is chest X-ray interpreted ? Chest X-ray interpretation is critical for so many disease including pneumonia and lung cancer. Radiologist is trained on the interpretation of x-rays. What is radiologist ? radiologist is a medical doctor who specializes in diagnosing and treating disease and injury, using medical imaging techniques such as x-rays, computed tomography (CT scan), magnetic resonance imaging (MRI)



Let's look at one abnormality called a mass look like, it's a chest x-ray that contain a mass and the chest x - rays that are normal. A mass is defined as a damage of tissue seen on the chest x - ray as greater than 3 centimeters in diameter. The algorithm eventually learns to go from a chest x-ray to produce whether the x-ray contains a mass or not. The algorithm produces an output in the form of scores , which are probabilities that the image contains a mass(Hypothesis representation→p(y=1|x;theta)).

Evaluation metrics is expressed in to two ways

1) How much the model is good , this is called the fitness function 2) How much the model is bad at , this is called the cost function.



So the probability that this image contains a mass is outputted to be 0.48, and the probability for this image is outputted to be 0.51. When training has not started, these scores, these probability outputs are not going to match the desired label , because as we set the weights randomly at first we only can be right 10% at the first time.

Let's say the desired label for mass is 1, and for normal is 0. And you can see that 0.48 is far off from 1 and 0.51 is far off from the desired label of 0. And we can

measure this error by computing a loss function. A loss function measures the error between our output probability and the desired label. But as it learns it will produce scores that are closer to the desired labels over time.

Image Classification and Class Imbalance

There are basically key challenges for training algorithms on medical images, challenges which make training medical image classification algorithms more challenging. We'll talk about three key challenges for training algorithms on medical images; the class imbalance challenge, the multitask challenge and the datasets size challenge. For each challenge, we'll cover one or two techniques to tackle them.

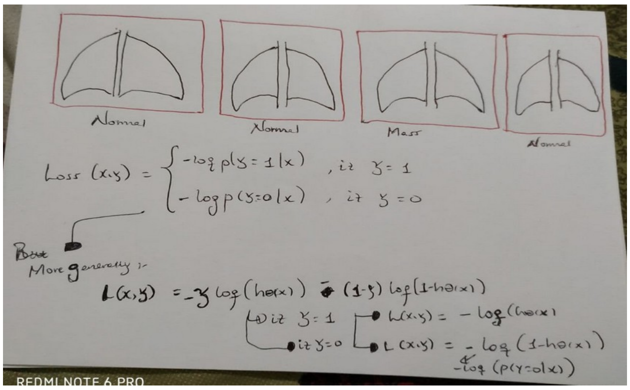
For the class imbalance problem we have a solution called weighted loss and sampling. For the multi class problem we have a solution called multi label loss. For the dataset size problem we have a solution called Transfer learning + Data augmentation.

Let's start with the class imbalance challenge. So here's the challenge, There's not an equal number of examples of non-disease and disease in medical datasets. Why ? This is a reflection of the prevalence or the frequency of disease in the real-world, where we see that there are a lot more examples of normals than of mass, especially if we're looking at X-rays of a healthy population. In a medical datasets, you might see 100 times as many normal examples as mass examples.

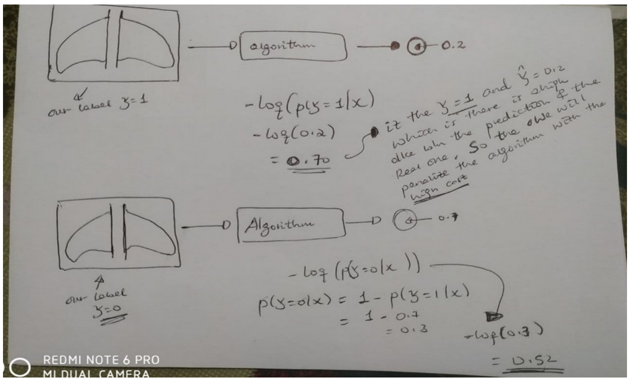
Class Imbalance

- ✔ What does it mean by class imbalance ?
- ✔ Why class imbalancing problem is happening ?
- ✔ How to calculate the Loss function for binary and multi label problem ?
- ✔ What is the impact of class imbalance on loss calculation and on our project in general ?
- ✔ How to solve class Im-balancing problem for both binary and multi-label problem ?

Basically this happens when we take real world datasets in count.


Normal
Normal
Mass
Normal
Loss (x,y) = { -log p(y=1|x) , if y=1 ; -log p(y=0|x) , if y=0
~~But~~
More generally :-
L(x,y) = -y log(hθ(x)) + (1-y) log(1-hθ(x))
if y=1 → L(x,y) = - log(hθ(x))
if y=0 → L(x,y) = - log(1-hθ(x))
-log (p(y=0|x))
REDMI NOTE 6 PRO

The loss over here is called binary cross entropy loss and this measures the performance of a classification model whose output is between 0 and 1.

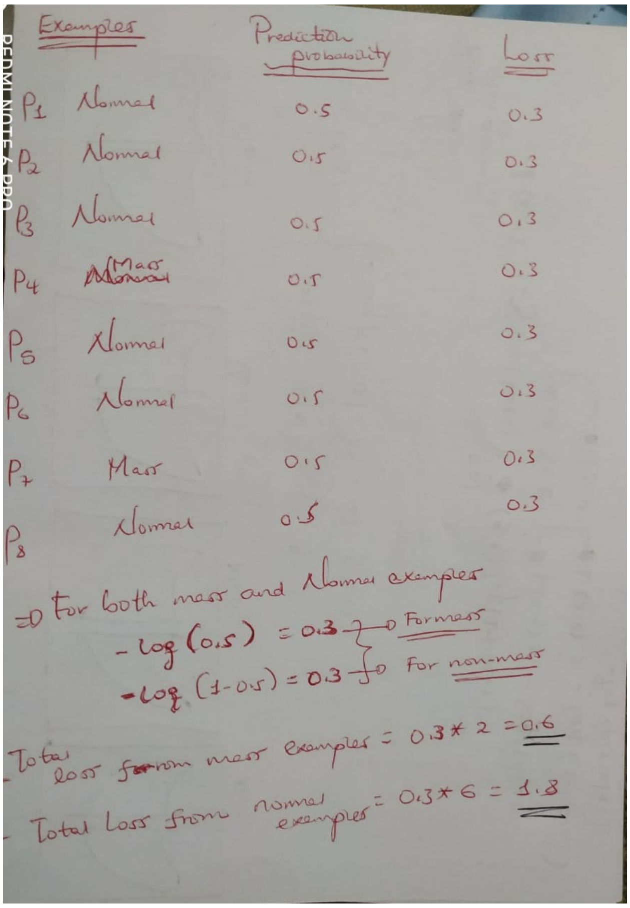


Now in the above image the first train set has a probability of 0.2 , this is the probability that this example is a mass. p(y=1|x). so we can apply a loss to compute the loss on our example. Now notice that our desired label is 1. so we are going to use the first formula.

If we see the second train set example this time our desired label is a non - mass (normal) which means has a label of 0. so we are going to use the second formula.

Impact of class imbalance on Loss Calculation

Let's say when training hasn't started let's say the algorithm produces an output probability of 0.5 for all of the examples.



REDMI NOTE 6 PRO

Examples		Prediction probability	Loss
P1	Normal	0.5	0.3
P2	Normal	0.5	0.3
P3	Normal	0.5	0.3
P4	~~Normal~~ Mass	0.5	0.3
P5	Normal	0.5	0.3
P6	Normal	0.5	0.3
P7	Mass	0.5	0.3
P8	Normal	0.5	0.3

⇒ For both mass and Normal examples

$-\log(0.5) = 0.3 \rightarrow$ For mass

$-\log(1-0.5) = 0.3 \rightarrow$ For non-mass

- Total loss from mass examples = 0.3 * 2 = 0.6
- Total Loss from normal examples = 0.3 * 6 = 1.8

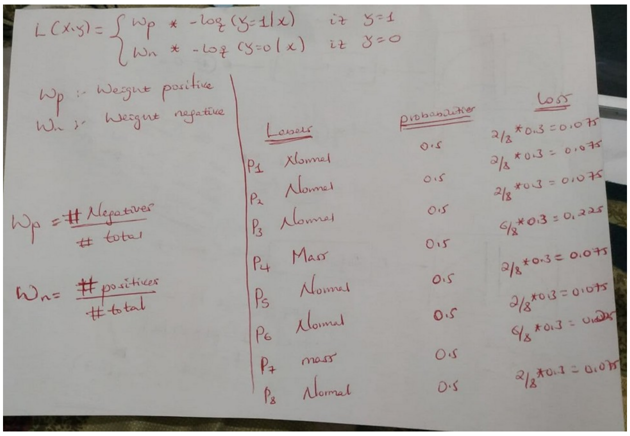
Notice how most of the contributions to the loss is coming from a normal examples rather than from a mass examples , so the algorithm is optimizing it's update to get normal examples and not giving weight to the mass examples. This is a class imbalance problem.

The solution to the class imbalance problem is to modify the loss function to weight the normal and mass classes differently.

Weighted Losses

- ➢ What are weighted losses ?
- ➢ Why do we need them ?
- ➢ How are they going to solve our problem ?

The solution to the class imbalance problem is to modify the loss function to weight the normal and mass classes differently.



$$L(x,y) = \begin{cases} W_p * -\log(y=1|x) & \text{if } y=1 \\ W_n * -\log(y=0|x) & \text{if } y=0 \end{cases}$$

Wp :- Weight positive

Wn :- Weight negative

$W_p = \frac{\# \text{Negatives}}{\# \text{total}}$

$W_n = \frac{\# \text{positives}}{\# \text{total}}$

	Labels	probabilities	Loss
P1	Normal	0.5	2/8 * 0.3 = 0.075
P2	Normal	0.5	2/8 * 0.3 = 0.075
P3	Normal	0.5	2/8 * 0.3 = 0.075
P4	Mass	0.5	6/8 * 0.3 = 0.225
P5	Normal	0.5	2/8 * 0.3 = 0.075
P6	Normal	0.5	2/8 * 0.3 = 0.075
P7	mass	0.5	6/8 * 0.3 = 0.225
P8	Normal	0.5	2/8 * 0.3 = 0.075

Wp :- will be the weight we assign to the positive or to mass examples and Wn :- will be to the negative or normal examples. Now as we saw what happens is we weight the positive examples more so that they will have equal contribution to the overall loss.

In general case , the weight we'll put on a negative class will be the number of positive examples over the total examples. The weight we will put on the positive class will be the number of negative examples over the total examples.

- The total loss of the positive class is :-

  2 * 0.225 = 0.45//

- The total loss of the negative class is :-

  0.075 * 6 = 0.45 //

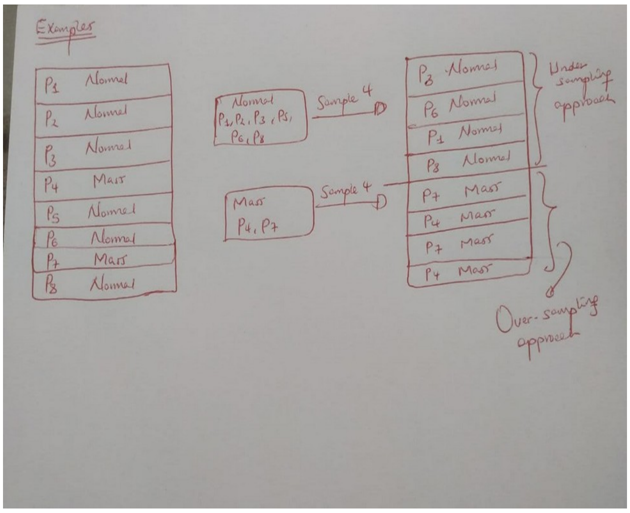
So this shows that we have equal contribution to both of the classes , this process is called weighted loss.

Resampling to achieve Balanced Classes

- What does it mean by resampling ?
- Why do i need to use this method ?
- How does Resampling solve the problem of class imbalance problem ?

The basic idea here is to resample the dataset such that we have an equal number of normal and mass examples.

First we group the normal and the mass examples together , notice that the normal group has six examples and the mass has two examples.



Now from these two groups we will sample the images such that there's an equal number of positive or mass class and half of the examples from the negative or normal class , this means we may not able to include all of our normal examples in our re sample further more we may have more than one copy of the mass examples in our re-sampled datasets. This method is sometimes called over-sampling and under-sampling Approach.

- Question :- Don't we lost some information when we under sample it ?

Now , if we calculate the loss :- there will be equal distribution to the loss from the normal examples and from the mass examples.

Total loss from mass examples :-

0.3 * 4 = 1.2

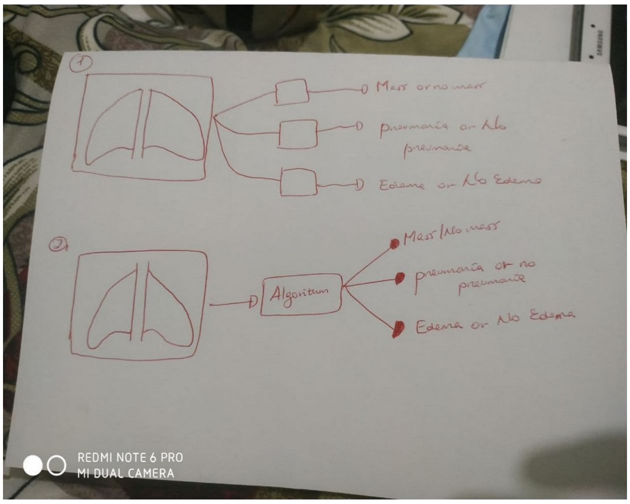
Total loss from Normal examples :-

0.3 * 4 = 1.2

Multi Task Learning

- ✔ What does it mean by multi task learning
- ✔ Why do we need this learning method
- ✔ How do we calculate the loss for multi task learning
- ✔ How do we solve the class imbalance in multitask problems

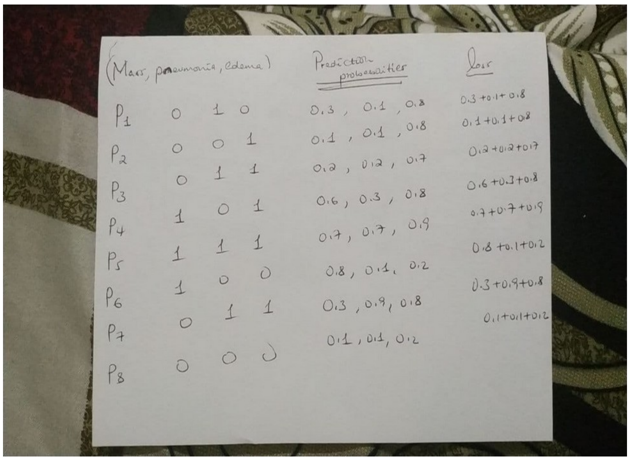
By far we've looked at binary classification , where we care about classifying whether an example is a mass or not a mass However in the real world we care about classifying the presence or absence of many such disease.



One simple way to do multi label classification is to have models that each learn neural network one of these tasks how ever this approach is not good because for each and every tasks we are trying to learn it from the scratch. But instead another way to do multi label classification is to have one model which computes all of the classifying , these allows us to learn common features in all of the tasks which allows us to use our existing data more efficiently.

What is edema :- It’s an excessive fluid in lung

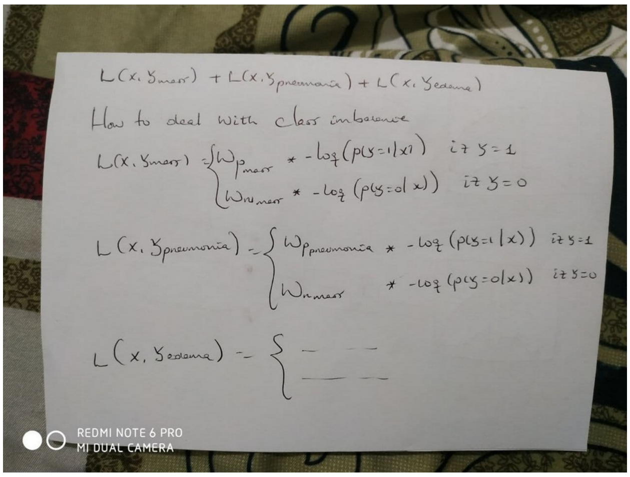
Example :-



(Mass, pneumonia, edema)				Prediction probabilities	Loss
P1	0	1	0	0.3 , 0.1 , 0.8	0.3 +0.1+ 0.8
P2	0	0	1	0.1 , 0.1 , 0.8	0.1 +0.1+ 0.8
P3	0	1	1	0.2 , 0.2 , 0.7	0.2 +0.2+0.7
P4	1	0	1	0.6 , 0.3 , 0.8	0.6 +0.3+0.8
P5	1	1	1	0.7 , 0.7 , 0.9	0.7 +0.7 +0.9
P6	1	0	0	0.8 , 0.1 , 0.2	0.8 +0.1+0.2
P7	0	1	1	0.3 , 0.9 , 0.8	0.3 +0.9+0.8
P8	0	0	0	0.1 , 0.1 , 0.2	0.1+0.1+0.2

So instead of having a one label , they now have one label for every disease in the example , where zero denotes the absence of that disease and one denotes the presence of that disease and instead of outputting one output from the model , the model now has three different output the probability of different disease.

So now for the multi-label case , we need a modification to our loss function.



$L(x, y_{mass}) + L(x, y_{pneumonia}) + L(x, y_{edema})$

How to deal with class imbalance

$$L(x, y_{mass}) = \begin{cases} W_{p_{mass}} * -\log(p(y=1|x)) & \text{if } y=1 \\ W_{n_{mass}} * -\log(p(y=0|x)) & \text{if } y=0 \end{cases}$$

$$L(x, y_{pneumonia}) = \begin{cases} W_{p_{pneumonia}} * -\log(p(y=1|x)) & \text{if } y=1 \\ W_{n_{mass}} * -\log(p(y=0|x)) & \text{if } y=0 \end{cases}$$

$$L(x, y_{edema}) = \begin{cases} \text{—— ——} \\ \text{—— ——} \end{cases}$$

we modified the loss function such that we look at the error associated with each disease.

Model Tasting

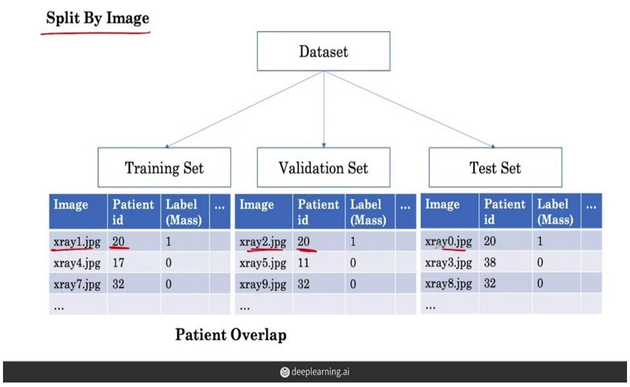
- ✗ How to test the model ?
- ✗ What is patient overlapping ?
- ✗ What is the problem with patient overlapping ?
- ✗ How can we solve patient overlapping

Let’s say the patient comes in twice for an x-ray once in June and once in November. Both times the patient wears a necklace when he/she has the x-ray taken. One of the x-ray is sampled as part of the training set and other as part of a test set.

We train our deep learning model and find that it correctly predicts normal for the x-ray in the test set. The problem is that it is possible that the model actually memorized to output normal when it saw the patient when a necklace is on. This is not hypothetical deep learning models can unintentionally memorize training data and the model could memorize rare or unique training data aspects of the patient. This would lead to an overly optimistic test set performance.

To tackle this problem in our dataset we can make sure that a patients x-rays only occur in one of the sets. Now , if the model memorizes the necklace on the patient , it does not help it to achieve a higher performance on the test set because it does see the same patient.

When we split a dataset in the traditional way , images are randomly assigned to one of the sets.



Training Set

Image	Patient id	Label (Mass)	...
xray1.jpg	20	1	
xray4.jpg	17	0	
xray7.jpg	32	0	
...			

Validation Set

Image	Patient id	Label (Mass)	...
xray2.jpg	20	1	
xray5.jpg	11	0	
xray9.jpg	32	0	
...			

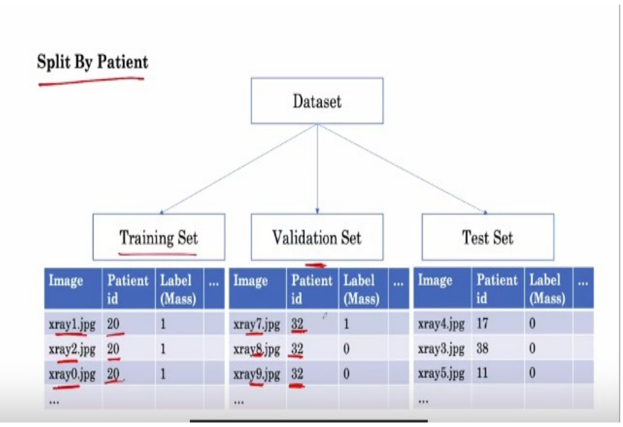
Test Set

Image	Patient id	Label (Mass)	...
xray0.jpg	20	1	
xray3.jpg	38	0	
xray8.jpg	32	0	
...			

Patient Overlap

deeplearning.ai

This is a patient overlap. Now notice that this way x-rays that belong to the same patient in different sets. For instance X-ray one belongs to a patient id 20 and it’s part of the training , it’s also belongs in the validation , also in the test set. This is the problem of patient overlap instead we split a dataset by a patient , all of the x-rays that belong to the same patient are in the same set.



Training Set

Image	Patient id	Label (Mass)	...
xray1.jpg	20	1	
xray2.jpg	20	1	
xray0.jpg	20	1	
...			

Validation Set

Image	Patient id	Label (Mass)	...
xray7.jpg	32	1	
xray8.jpg	32	0	
xray9.jpg	32	0	
...			

Test Set

Image	Patient id	Label (Mass)	...
xray4.jpg	17	0	
xray3.jpg	38	0	
xray5.jpg	11	0	
...			

So in the example , x ray 1 , x ray 2 , x ray 3 , which all belong to patient 20 are all part of the training and seven , eight , nine which are all in the validation set and we don’t see 20 and 32 here in our test set. This way we can make sure that there is no any patient overlap.

Ground Truth and Consensus Voting

- ➔ What is a ground truth ?
- ➔ How do i determine the correct label ?
- ➔ What are the different ways of finding the correct labels ?
- ➔ What is a consensus voting ?
- ➔ What is definitive test ?

One major question in testing a model is how we determine the correct label for an example. The right label is more commonly called the ground truth in the context of machine learning or the reference standard in the context of medicine.

So in order to determine the correct label we can use consensus voting , the ideas is a group of human experts to determine the ground truth. Let’s say we have a three radiologists look at the chest x-ray and each determine whether there is pneumonia or not. In general the answer will be the majority vote of the three radiology. Alternatively we can have the three radiologist get in to a room and discuss their interpretation until they reach a single decision which can then be used as the ground truth.

The second method is to use a more definitive test which provides additional information to set the ground truth. For example :- to determine whether a patient has a mass using a chest x-ray a more definitive test that can be performed is CT-Scan. This gives additional information for the **radiologist.**

We have looked at two methods to set the ground truth , which are consensus voting and having more definitive test , with the second method the difficulty is that we might not have these additional test available not every one who gets a chest x-ray gets a CT scan.

Medical Image Segmentation

- ➢ What does it mean by Image segmentation ?
- ➢ What does it mean by Medical Image Segmentation ?
- ➢ What is the application of studying medical image segmentation ?

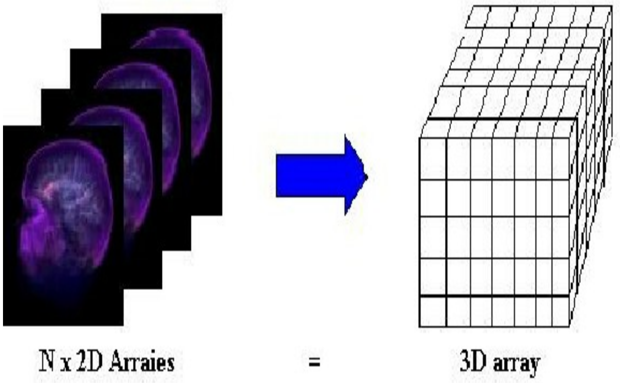
Until now we have been classifying whether or not a certain image has a disease or not but using Image segmentation we can also tell about which part of the image contained the disease.

MRI Data and Image Registration

- ✔ What is MRI stands for (Magnetic Resonance Image)
- ✔ What is the exact look of an MRI Data
- ✔ How to represent MRI Data’s to make them input for the Segmentation model ?
- ✔ What is Image Registration ?
- ✔ Why do i need to register our image ?

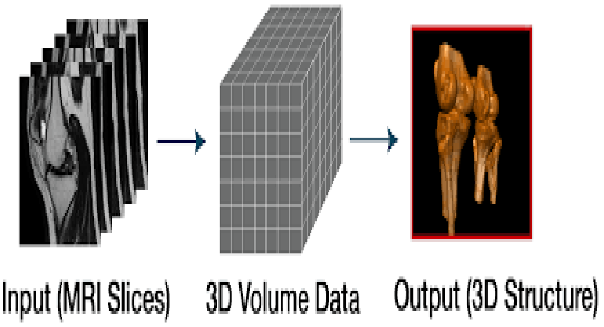
We will learn how we can train and evaluate our segmentation model for identifying tumors on MRI data , we’ll start by talking about representing MRI data. *why does it mean by representing ?* it’s just a process of converting an MRI data to numbers to be used as an input by the segmentation model. Why do we chose segmentation model for MRI problems ?

As we’ve seen , our MRI images are not just a single 2D image like x-rays but instead, also MRI sequence is a 3D Volume. *Show that and if it a 3D what are the components ?*



An MRI example will be made up of multiple sequences and this will consist of multiple 3D volumes. The key idea that we will use to combine the information from different sequences is to treat them as different channels. we can say one of the channels is red , one is green and one is blue channel in the same way we have three channels of an RGB image.

The analogy of three channels representing the RGB channels is mostly useful for us to visualize how we're going to combine the channels. There is nothing special about the number three. The idea of using different channels is also extends to when we may have four or five sequences, which can be represented with four or five channels. Now we will combine each of the sequences together to produce one image which is the combination of all images. what are these channels represent ?



So to the machine these channels are stacked in depth dimension. One challenge with combining these sequences is that they may not be aligned with each other. For instance , if a patient moves between the acquiring of each of these sequences , their head might be tilted in one sequence compared with the others. If the images are not aligned with each other when we combine them , the brain region at one location in the red channel does not correspond to the same location in the green or the blue channels . *So how do we fix this problem ?* A pre-processing approach that is often used to fix this is called image registration. The basic idea with image registration is to transform the images so that they are aligned or registered to each other. *How does this image registration works ?*

Segmentation

- ➢ What is segmentation ?
- ➢ Why do i need this model ?
- ➢ How does this segmentation model help us for MRI data ?

Once the image has been registered, we're able to combine different sequences. Now that we've seen how we can represent MRI data , let's turn to the task of brain segmentation. In this case , we're trying to define the class of every point in the 3D volume. These point in 2D spaces are called pixels.

Let's discuss the two approaches to segmentation with MRI data. The first is 2D approach and the second is 3D approach. In the 2D approach we break up the 3D MRI volume we've built in to 2 many 2D slices. each one of these slices passed in to a segmentation model which outputs the segmentation for that slice. One by one , each slice is passed through the segmentation model in this manner to generate a segmentation for every slice. Then the 2D slices can be then combined once again to form the 3D output volume of the segmentation.

The drawback with this 2D approach is that we might lose important 3D context when using this approach. For instance if there is a tumor in once slice , there is likely to be a tumor in the slice right adjacent to it , since we are passing in slices one at a time in the network , the network not able to learn this useful context. *See how this happens in visualization ?*

In the 3D approach , ideally we'd want to pass the whole MRI volume in to segmentation model and get out a 3D segmentation map for the whole MRI.

However the size of the MRI volume makes it impossible to pass it in all at once in to the model , it would simply take two much memory and computation. *So what to do ?* what we can do instead is to still have the model to be able to get this cortex information in the depth dimension. *Can we do this approach in 3D volume ?* In the 3D approach we break up the 3D MRI volume in to many 3D sub volumes. Each of these sub volumes has some width , height and depth context. So like in the 2D approach , we can feed the sub volumes now one at a time in to the model and then aggregate them at the end to form a segmentation map for the whole volume.

2D U-Net and 3D U-Net

- ✔ What are this architectures ?
- ✔ How does they work ?
- ✔ What does the name U represents ?

Now that we've covered 2D and 3D approaches to segmentation. Let's divide in to segmentation architectures. One of the most popular architectures for segmentation has been the U-Net. The U-Net architecture owes it's name to a U-like shape.

The U-Net consists two paths , a contracting path and an expanding path.