WEEK 1:

Medical Image Diagnosis

(1:41) Which of the following best describes how a model can be trained to recognize a cancerous image using a Convolutional Neural Network (CNN)? -- Intuition, single answer

- Train the model by loading many images labeled as either cancerous or non-cancerous.
  - Correct!
- Train the model by loading a single cancerous image and non-cancerous image.
  - Try again. If a single image is uploaded from both categories, what happens if the computer is provided an image that is nothing alike to either of these categories?
- Tell the model explicitly how a cancerous image differs from a non-cancerous image.
  - Try again. We are trying to train the model to recognize a cancerous image on its own. Thus we cannot explicitly tell the model what the image is every time, especially if we load many images.
- Mark all images as cancerous, then repeatedly make the computer try to pick non-cancerous images.
  - We already have labeled the images as cancerous, which means that the model using a CNN would learn that these images are cancerous, when they may not necessarily be the case.

Eye Disease and Cancer Diagnosis

(none, question from Medical Image Diagnosis applies very closely with this video)

Building and Training a Model for Medical Diagnosis

(2:30) We were able to detect from the chest X-rays that masses could be detected using other X-ray images that had a mass in it. Which of the following binary classifications will best diagnose if a patient has a mass? -- Intuition, single answer

- Mass / No mass
  - Correct! This classification will clearly define whether or not a patient has a mass from their X-ray scan.
- Mass on Left / Mass on Right / No mass
  - Try again, while it is possible to have more than two different classifications, the question asks which “binary” classification will best diagnose the patient. Thus, do we need to know where the mass is?
- Small mass / Large mass
  - Try again, Pranav defined a mass to be 3 centimeters or greater in diameter. Thus, if a mass is “small”, are we saying that it is smaller than or greater than 3 centimeters? If we define it to be larger, then a small mass and a large mass are

essentially classified as the same thing. If we classify it as “small” then either the mass isn’t defined as a mass (less than 3 centimeters), or we don’t have a parameter to distinguish small from large.

- Normal shape / Irregular shape
  - Try again, the problem with this classification is that the “shape” is not clearly defined. This could be the shape of a person’s body or even the shape of the mass. On top of this, the mass can take many different shapes and locations, which means that the training of the CNN would involve trying to figure out what is a “normal shape”. There is a much simpler option than this.

Training, prediction, and loss

(1:52) If our model is training to recognize a mass from a chest X-ray and the predicted label is incorrect, then we have a bad training model. -- Intuition, True / False

- True
  - Try again, as we train the model more and more, the goal is to achieve an error that is decreasing over time. This is perhaps the most likely way to determine whether or not we have a good or bad training model. Just like in real life, humans are likely to not get something correct if they are learning it for the first time! Think about training models this way.
- False
  - Correct! In fact, during training, we don’t necessarily want to have 100% accuracy in guessing correctly due to a concept known as overfitting. You will learn about this concept later in this specialization.

Image Classification and Class Imbalance

(none, question probably ties in best with Impact of Class Imbalance on Loss Calculation video).

Binary Cross Entropy Loss Function

(none, pretty straightforward)

Impact of Class Imbalance on Loss Calculation

(3:18) Why is it important to multiply a positive / negative weight to the Binary Cross-Entropy Loss function when dealing with Class Imbalance? -- Intuition, single answer

- If we have an unequal number of examples between two classes, then the group with more training examples is more likely to be adjusted throughout training, thus being more likely to be correct.
  - Correct! This is how we can adjust the total loss of two groups to be equal.

- If we have an unequal number of examples between two classes, then the group with fewer training examples is more likely to be adjusted throughout training, thus being more likely to be correct.
    - Try again, the weights are trying to compensate for the class that has fewer training examples. Recall that the weight that is multiplied to the Binary Cross-Entropy Loss function will be the number of examples in the opposite class divided by the total number examples. If we have only 2 mass examples out of 8 total examples, that means we are multiplying the loss function by 6/8. How will this compare to the total loss of the other examples?
- Multiplying a weight to the Binary Cross-Entropy loss function has no effect on Class Imbalance.
    - Try again, this really only happens if we have an equal number of examples from both classes. This means that we wouldn't even need to multiply a weight, and we wouldn't even have a Class Imbalance.

Resampling to Achieve Balanced Classes

(1:39) Which of the following resampling techniques helps to achieve the Class Imbalance problem? -- Intuition, multiple answer

- Undersampling the class with less training examples
    - This should not be selected. By undersampling a class with fewer training examples, we are essentially removing more examples from a class that we need more training examples from relative to the other class. This means that the total loss from this class is going to be even smaller compared to the other class, thus creating a larger Class Imbalance.
- Undersampling the class with more training examples
    - Correct! Doing this may cause some examples from this class to not be included in the training, but will prevent the total loss from this class from being larger than the total loss of the other class.
- Oversampling the class with less training examples
    - Correct! Doing this may cause some examples from this class to be duplicated for training, but will prevent the total loss from this class from being smaller than the total loss of the other class.
- Oversampling the class with more training examples
    - This should not be selected. By oversampling a class with more training examples, we are essentially including more examples from a class that we need fewer training examples from relative to the other class. This means that the total loss from this class is going to be even larger compared to the other class, thus creating a larger Class Imbalance.

Multi-Task

Multi-tasking requires more data because the model needs to recognize different features that lead to multiple diseases.

- True
    - Try again, what makes multi-tasking so effective is that we can train the model to recognize multiple diseases with the same data. Thus, we have multiple labels per image to recognise all diseases that the model can train on. While training may differ and possibly slow down, we don't necessarily need "more" data.
- False
    - Correct!

Multi-task Loss, Dataset size, and CNN Architectures

(none)

Working with a Small Training Set

(2:54) What benefit does Transfer Learning provide for our model if we are trying to identify diseases from a chest X-Ray? -- Intuitive, single answer

- By transferring the predictions from another ConvNet with the same data, we save time training our model because the model does not need to retrain on the same data.
    - Try again, we aren't copying the predictions made from the same data of another model. Think about what a ConvNet actually does and how a ConvNet helps to train the model. What does a ConvNet identify (see Multi-task Loss, Dataset size, and CNN Architectures).
- By transferring the learned features from a ConvNet, we can save time on our model by copying the learned features, then fine-tuning our model to help us identify diseases in X-rays.
    - Correct! We can pretrain our model from another ConvNet to give ourselves a starting point in our training. We can fine-tune the model to adjust any irregularities or errors, but overall, we do save time across training the entire data set.
- Transfer Learning allows our model to transfer labels to different images, which saves time by copying the same labels from the previous data.
    - Try again, changing the labels for different images is generally a bad idea, even if it could save time. We most likely will have a larger Total loss, which may take longer for our model to train.

Generating More Samples

Data Augmentation doesn't actually give our model more data, but rather modifies the data to allow our model to think we have more training examples.

- True
    - Correct! This is important because acquiring data can be time consuming and sometimes even costly. Thus, we can train the model with the same amount of data that we have but creating more examples per X-ray (or piece of data) that we have.
- False
    - Try again, we actually are not collecting more data! Collecting data can sometimes be costly or time consuming, which is why data augmentation is a useful tool if we don’t have enough data to provide for our model. Think about data as the actual X-rays themselves. We aren’t collecting more physical X-rays to give to our model. Data augmentation allows us to provide more examples for the model using the same amount of data.

WEEK 2

Sensitivity, Specificity, and Evaluation Metrics

(none, pretty straightforward and clear)

Accuracy in terms of conditional probability

(1:41)Which of the following best describes accuracy of a model? -- Intuitive, multiple answers

- Number of examples correctly classified / total examples
  - Correct! This answer is more generalized mathematically, but still shows accuracy for the model.
- The ratio of correctly classified examples to incorrect examples
  - This should not be selected. Accuracy is a metric that usually ranges from 0 to 1, where 0 is complete inaccuracy and 1 is complete accuracy. If we have more correctly classified examples than incorrect examples, then our accuracy would be greater than 1. Additionally, having an 0 incorrectly classified examples would lead to an accuracy value that doesn't exist. Think about how we can keep accuracy between 0 and 1.
- The sum of the probability that the model correctly predicts a patient with disease and the probability of model correctly predicting a patient that is normal.
  - Correct! This is accuracy in terms of conditional probability. This can also be rewritten as P(A | B) P(B), or more specifically, P(correct | disease) P(disease) + P(correct | normal) P(normal)
- The sum of correctly classified examples and incorrectly classified examples.
  - This should not be selected. The sum of correctly classified examples and incorrectly classified examples will give the total number of training examples. If accuracy was defined as this, accuracy would always be equal to the number of training examples, rather than being between 0 and 1. Think about how we can keep the accuracy between these two values.

Sensitivity, Specificity and Prevalence

(4:08) In terms of calculating the accuracy, which of the following is the most correct definition of sensitivity, specificity, and prevalence? -- Intuitive, single answer

- Sensitivity and specificity are numerical weights, while the prevalence is the frequency of a particular event happening.
  - Try again. Think about the prevalence. How do we calculate prevalence? If the prevalence was the frequency, then we could have an accuracy that is greater than 1. What must the prevalence (and (1 - prevalence)) be equal to in order to range between 0 and 1 when multiplied by the sensitivity and specificity?
- Sensitivity and specificity are numerical weights, while the prevalence is the probability of a particular event happening.

    - Correct! Sensitivity and specificity will be multiplied by a value that is less than 1 (which is the prevalence, as well as (1 - prevalence)). When this happens, we get two values that we need to sum together in order to get that accuracy.
- Sensitivity is a numerical weight, specificity is the frequency of a particular disease happening, and the prevalence is the value that divides accuracy.
    - Try again. Sensitivity and specificity are more related to each other than a numerical weight and frequency. If prevalence divided accuracy, then if we multiplied the sensitivity and specificity by the prevalence, we would not achieve accuracy because we need to consider (1 - prevalence) as well.
- None of the above
    - Try again. Think about what sensitivity and specificity actually are. Are they closely related? How can we use prevalence to calculate the accuracy if we put accuracy in terms of these three equations?

PPV, NPV

(2:12) PPV + sensitivity is always equal to 1, as well as NPV + specificity is always equal to 1. -- Intuitive, True/False

- True
    - Try again. Sensitivity is calculated by taking the number of examples that are classified correctly for a disease, and dividing that value by the number of examples with a disease. The PPV is calculated by taking the number of examples that are classified correctly for a disease and dividing it by the number of examples that the model classifies as a disease. The difference between the PPV and sensitivity (as well as NPV and specificity) is that the PPV (and NPV) compare correctly classified examples to the Ground Truth, while the sensitivity and specificity compare correctly classified examples to the Model's predictions. Thus, we cannot say that the PPV and sensitivity is always equal to 1.
- False
    - Correct! While similar and even related, the sum of the PPV and sensitivity do not add to equal 1.

Confusion matrix

(2:16) Suppose that I have a model that I want to train to recognize fruit. The model is being trained such that it will use binary classification to recognize an image as a piece of fruit (marked as positive, or with a 1), or not a piece of fruit (marked with negative, or 0). Suppose I load 20 images into the model, where 13 of the images are fruit, and 7 are not. Out of the 13 images that are real fruit, the model predicts that 5 of the images are fruit. Out of the 7 images that are not fruit, the model predicts that 6 of the images are not fruit. Given this information, calculate the NPV of this model (NPV = TN / (TN + FN)). -- Intuitive, single answer

- 5/13
  - Try again. This value is the sensitivity (TP / (TP + FN)). Recall that the NPV = TN / (TN + FN). Given the information, how can we find TN and FN? Try to make a confusion matrix with the information above. Hint: We can find the FN by finding the number of fruit images that are incorrectly classified as non-fruits. In other words, how many fruit images did the model predict to be a non-fruit?
- 6/7
  - Try again. This value is the specificity (TN / (FP + TN)). Recall that the NPV = TN / (TN + FN). Given the information, how can we find TN and FN? Try to make a confusion matrix with the information above. Hint: We know that the TN from this scenario is equal to 6. Fill out the rest of the confusion matrix.
- 3/7
  - Correct! We can see that the True Negatives, or images of non-fruits that are correctly classified as non-fruits, is equal to 6. The False Negatives, or images of fruits that are incorrectly classified as non-fruits, is equal to 8. Thus, NPV = 6 / (6 + 8) = 6/14 = 3/7.
- 5/6
  - Try again. This value is the PPV (TP / (TP + FP)). Recall that the NPV = TN / (TN + FN). Given the information, how can we find TN and FN? Try to make a confusion matrix with the information above. Hint: We can find the FN by finding the number of fruit images that are incorrectly classified as non-fruits. In other words, how many fruit images did the model predict to be a non-fruit?

ROC curve and threshold

(none, next video)

Varying the threshold

(2:40) Suppose we have a set of 20 training examples where an example of each class alternates consecutively on the score line(disease, normal, disease, normal, etc.). The spacing of each training example and their score is evenly distributed. Assume our threshold starts at t = 0.5. How would you expect the sensitivity and specificity to change as we decreased the threshold to t = 0.25? -- Intuitive, single answer

- Sensitivity and specificity both begin at 1/2. When t = 0.25, sensitivity increases approximately to 3/4, while specificity decreases to 1/4.
    - Try again. What happened when we saw the threshold at t = 0, and t = 1? Did sensitivity or specificity = 1 during these thresholds?
- Sensitivity and specificity both begin at 1/2. When t = 0.25, specificity increases approximately to 3/4, while sensitivity decreases to 1/4.
    - Correct! Generally speaking, specificity will increase as the threshold decreases, while sensitivity will increase.
- Sensitivity starts at 0, while specificity starts at 1. As t = 0.25, the specificity will decrease to approximately 1/2, while sensitivity will remain at 0.
    - Try again. Recall that sensitivity is the probability of the number of examples classified correctly by the model divided by the total number of diseased patients. How does the threshold increase or decrease either the sensitivity, specificity, or both?
- Sensitivity starts at 0, while specificity starts at 1. As t = 0.25, the sensitivity will increase to approximately 1/2, while sensitivity will remain at 1.
    - Try again. Recall that sensitivity is the probability of the number of examples classified correctly by the model divided by the total number of diseased patients. How does the threshold increase or decrease either the sensitivity, specificity, or both?

Sampling from the Total Population

(none, next video)

Confidence intervals

(2:45) The example in the video used a 95% confidence level to show the population accuracy to be in [0.72, 0.88] However, if we use a 100% confidence level, we will find the exact population accuracy because we are 100% confident. -- Intuitive, True / False

- True
    - Try again. While a 100% confidence level will include the population accuracy in 100% of samples, the range of the confidence interval will expand. If we have 95% confidence that the population that the population accuracy is in [0.72, 0.88], then we could technically also say that we are 100% confident that the population accuracy is in [0, 1]. If we want to find a more precise answer, we actually need to lower the confidence level in order to make a more defined boundary. However, doing so will risk the population accuracy to be more likely to be outside of the bounds.
- False
    - Correct! 100% confidence level means that we are more confident that the population accuracy is in between a broader interval, but doesn't really tell us much in finding precisely what the population accuracy is.

95% Confidence interval

(none, only talks about sample size variation)

WEEK 3:

Medical Image Segmentation

(none)

MRI Data and Image Registration

(3:26) Which of the following sequences (from first to last) best describes how we need to treat the MRI data in order to input this data into a segmentation model? You may assume that the MRI data takes the form of multiple 3D volumes seen in the Coronal view. -- Intuitive, single answer

- Image Registration, slice the 3D volumes, define the channels for the slices, combine the channels
    - Try again. Image Registration allows us to fix any alignment issues that we have in our channels. Thus, it would not make sense for Image Registration to be the first step. We need at least to have channels that we can combine in order to use Image Registration.
- Define channels for each slices, combine the channels, slice the 3D volumes, Image Registration
    - Try again. Think about what exactly we are treating as channels. Where did these channels come from? What did we do to the MRI data first?
- Slice the 3D volumes, combine the channels, Image Registration, define channels for each slices
    - Try again. Think about the second step and last step. How can we combine channels if they aren't clearly defined? What feature did we talk about in the video that defined channels for each slice? (answer: RGB values).
- Slice the 3D volumes, define channels for each slice, Image Registration, combine the channels
    - Correct! This is the correct sequence of events that will allow us to input MRI data into a segmentation model.

Segmentation

(3:40) Which of the following segmentation techniques can we use to help define the boundaries of a brain tumor? What is the drawback of each technique? -- Intuitive, multiple answers

- 2D Approach, but we may lose some 3D context when combining all of our slices.
    - Correct! A tumor in one slice will likely be present in other slices nearby or adjacent to this slice. However, since we pass in slices one at a time, the segmentation model won't detect that a tumor is present in nearby slices relative to a slice. The model would just be checking every single slice for a tumor.
- 2D Approach, but the segmentation model can't handle slices of this size.

    - This should not be selected. The segmentation model can actually handle 2D slices of the 3D volume. However, the segmentation model cannot handle the 3D volume itself.
- 3D Approach, but the segmentation model won't output a 3D volume of the same dimensions as the input.
    - This should not be selected. A segmentation model should return all of the 3D sub volumes that would make up the original 3D volume. When all of the sub volumes are combined after being passed through the segmentation model, we should be able to see the tumor's location or if it's present.
- 3D Approach, but we may lose some spatial context when combining all sub volumes.
    - Correct! If there is a tumor in one sub volume, there is likely to be another tumor in another sub volume as well. Or, there could be a tumor that is in between two sub volumes. Regardless, we still are losing this context because we are passing in sub volumes one at a time into the segmentation model.

2D U-Net and 3D U-Net

(2:56) 2D and 3D U-Nets behave very similarly, where one of their primary differences is that a 2D U-Net trains on 2D input/output slices, whereas a 3D U-Net trains on 3D input/output subvolumes. -- Intuitive, True/False

- True
    - Correct! U-Nets have a contracting and expanding path, as well as train on either 2D or 3D inputs/outputs, depending on the dimensions of the data
- False
    - Try again. U-Nets are "shaped" like a "U" because of the contracting path and expanding path. The main difference between the 2D and 3D net is how they train, and on what data they can input/output.

Data augmentation for segmentation

(none, i don't think it's necessary given how similar it is to week 1)

Loss function for image segmentation

(3:29) Compute the loss for the following set of data using the Soft Dice Loss function. -- Calculation, single answer

i	p	g
1	0.1	0
2	0.4	0
3	0.9	1

4	0.3	1
5	0.2	0

- 0.23
    - Correct!
- 0.77
    - Try again. Remember to subtract 1 by your calculated ratio.
- 0.29
    - Try again. Check your calculation with multiplying the sum of all (p * g) elements and dividing it by the sum of all p squared and g squared terms.
- 0.57
    - Try again. Check your calculation with multiplying the sum of all (p * g) elements and dividing it by the sum of all p squared and g squared terms.

Different Populations and Diagnostic Technology

(none)

External validation

Since our model has been trained for patients in the United States, the model may possibly be less accurate in other countries, such as Mexico. However, it is a good idea to completely retrain our model to become more accurate in Mexico, thus allowing us to have a good model for both the U.S. and Mexico.

- True
    - Try again. We shouldn't have to completely retrain the model in this case. The only case where we may have to do this is if we change our objective for using AI, or if our loss is gradually increasing with more training examples. What we can do to have more accuracy with examples from other countries is to fine-tune our model with other sets of data. This will allow our model to train on different examples which can support other countries' style of X-rays.
- False
    - Correct!

Measuring Patient outcomes

(none)