Deep Learning
Practice Questions

Subir Varma, Santa Clara University

1. **What are the main types of Deep Learning Systems. Give an example of each type (use new examples, not the ones given in class)?**

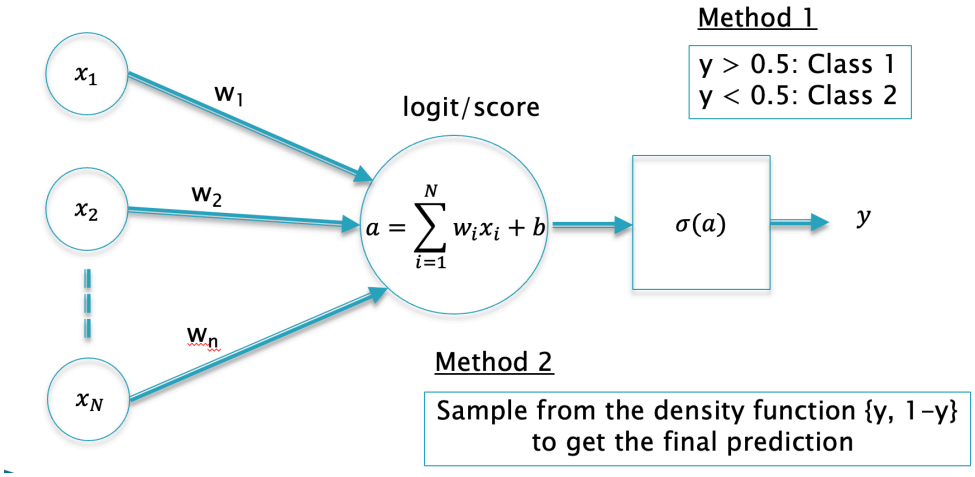
2. **How are Deep Learning systems different from older Machine Learning systems?**

3. **Given a Random Sequence (X1, X2, X3, X4), if P(X1) = 0.2, P(X2|X1) = 0.7, P(X3|X1,X2) = 0.35, P(X4|X1, X2,X3,X3) = 0.1, compute the joint probability (P(X1, X2, X3, X4). Which Probability Law did you use to calculate this?**

4. **There are 4 balls in an urn. Each ball is either red or black. You start by believing that the probabilities that the urn contains 0, 1, 2, 3, 4 red balls are all equal. You then reach into the urn and pull out a ball at random. It is red. Compute the new probabilities that the urn contains 0, 1, 2, 3 or 4 red balls.**

Linear Systems

5. **Refer to the figure below. For the case of two inputs (x1, x2), show that the condition y > 0.5 is equivalent to using a straight line to separate out the decision boundary.**



6. **Since Linear Systems can only classify using linear separators between categories, they cannot be used for datasets in which the separating boundary is more complex. Is it possible to get around this limitation (without resorting to deep networks)?**

7. **Why is it preferable to use Stochastic Gradient Descent or Batch Stochastic Gradient Descent, as opposed to plain Gradient Descent?**

8. **Why do Linear Systems that classify into two categories require only a single output? Would this still be the case if the Linear System was being used to do Regression rather than Classification?**

9. **Answer the following questions:**

   (a) Derive the expression $\frac{\partial y}{\partial x}$ for the derivative of the Sigmoid Function

   $$y = \frac{1}{1 + exp(-x)}$$

   (b) Derive the expression $\frac{\partial y_k}{\partial a_i}$ for the derivative of the Softmax Function (both for $i = k \; and \; i \neq k$)

   $$y_k = \frac{exp(a_k)}{\sum_{i=1}^{K} exp(a_i)}$$

   (c) Using the results of Part (b), show that the derivatives $\frac{\partial \mathcal{L}}{\partial a_k}$ and $\frac{\partial \mathcal{L}}{\partial w_{kj}}$ for K-ary classification problem are given by (see Page 49 of the Lecture 3 slides):

   $$\frac{\partial \mathcal{L}}{\partial a_k} = (y_k - t_k) \; and \; \frac{\partial \mathcal{L}}{\partial w_{kj}} = x_j (y_k - t_k)$$

   (d) Show that for the case when the number of categories K = 2, the Softmax output Y of the Logistic Regression system, is equivalent to a system in which Y is computed using the Logistic Sigmoid.

10. **Consider the Linear Model with the number of categories K = 2. Suppose that the Cross Entropy Loss Function was replaced by the Mean Square Error (MSE) Loss Function.**

    (a) Compute the gradient $\frac{\partial \mathcal{L}}{\partial w_i}$ for the MSE Loss Function.

    (b) Compare the expressions derived in Part (a) for the MSE gradient $\frac{\partial \mathcal{L}}{\partial w_i}$, with

the following expression for the gradient of the Cross Entropy Loss Function.

$$\frac{\partial \mathcal{L}}{\partial w_i} = (y - t)x_i$$

Based on this, explain why the Cross Entropy Loss Function is superior to the MSE Loss Function for doing Classification.

Dense Feed Forward Networks

11. **What is the Template Matching interpretation of a Linear Classifier? Why doesn't Linear Model based template matching work well in all cases?**

12. **A Dense Feed Forward network works by transforming the input representation from (x_1,x_2,…,x_N) to (z_1,z_2,…,z_N) in co-ordinate space (see Lecture 4, Page 23). What is a desirable property of the transformed representation when doing classification?**

13. **Why are Activations Functions an essential ingredient of Deep Learning networks?**

14. **Until recently the number of layers in Deep Learning systems was limited to 10-20 at most. What is the innovation that has made networks with hundreds of layers possible, and how does it do it?**

15. **Give at least 3 reasons why the Vanishing Gradient problem was a frequent occurrence in older Neural Networks.**

Backprop

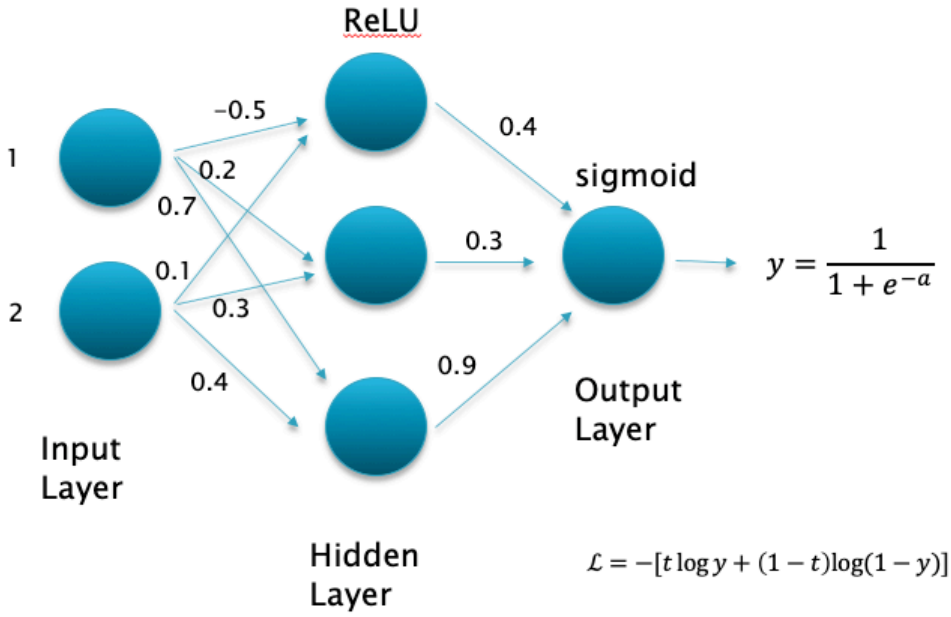
16. **How does the Backprop algorithm enable the training of deep Neural Networks?**

17. **Why is it not a good idea to use the sigmoid function as an Activation Function?**

18. **Assume that you are training a model with 1 million weights which takes up storage of X MB in your computer. How much additional storage do you need to perform a single pass of the Backprop algorithm (forward pass followed by backward pass)?**

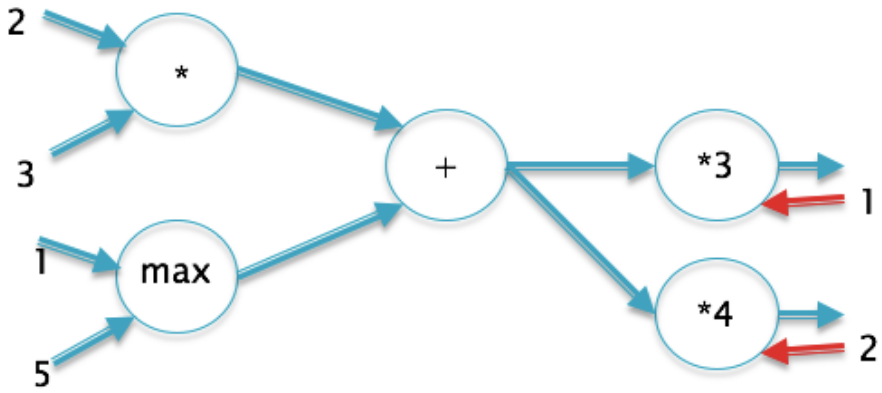
19. **Consider a Dense Feed Forward Network, of the type shown below, composed of an Input with 2 nodes, followed by a Hidden Layer with 3 nodes and finally an Output Layer with 1 node. Assume the activation function for the hidden layer is given by ReLU, the output activation is given by the sigmoid function**

and the Loss Function is the Binary Cross Entropy.



a. **Compute the number of parameters (weights and biases) required to describe the network.**
b. **Assume that the network is initialized with the weight values as given in the figure and the bias values are initialized to zero. If the input into the network is (1,2), compute the activations z_1, z_2, z_3 at the hidden layer nodes, and the output y of the network.**
c. **Assume that the output label t corresponding to the input (1,2) is 1. Recall that the gradient $\delta = \frac{\partial L}{\partial a}$ at the output node is (y-t). Backpropagate this gradient to compute the gradients $\delta_1, \delta_2, \delta_3$ at the three hidden nodes.**
d. **Based on the activations in Part (b) and the gradients in part (c), compute the gradients $\frac{\partial L}{\partial w}$ of the Loss Function with respect to the output weights (for the second layer of weights only).**

20. Consider the following computational graph:



Use the Gradient Flow Rules to compute the gradients for all the arcs in the graph.

Keras

21. **What kind of network topologies require the use of the Keras Functional API?**

22. **How are Callbacks used in Keras, give two examples to illustrate this.**

23. **When the training dataset is provided in raw form (jpg files or txt files), how does Keras figure out what label to assign to each sample in the dataset?**

Training Improvements

24. **State at least two benefits of using Stochastic Gradient Descent as opposed to plain Gradient Descent.**

25. **From the list below choose the reason(s) for the Vanishing Gradient problem:**
    (a) Use of the sigmoid activation function
    (b) Use of MSE Loss Function for classification
    (c) Bad initialization of model parameters
    (d) Very deep networks
    Is the answer (a), (b), (c), (d) or all of the above?

26. **What would happen if you were to use the Test Dataset to choose the best hyper-parameters in a Neural Network?**

27. **How does the Momentum Optimizer improve the convergence speed? Why does the RMSProp Optimizer work better than plain SGD?**

28. **Why is it not a good idea to initialize all the weights to the same value? (you can actually test this out by using the program you developed for solving Assignment 2, Problem 2).**

29. **Explain the concept of Model Capacity and the relationship between Model Capacity and Data complexity.**
    Why is it better to increase Model Capacity by adding layers to Deep Feed Forward Neural Network as opposed to adding more nodes per layer?

30. **If the Training Curve on Page 17 of Lecture 8 does not have the U shape but keeps decreasing, then what can you conclude. Can the addition of Regularization help solve this issue?**

31. **State two ways in which you can increase model capacity for the following types of networks:**
    (a) Linear
    (b) Dense Feed Forward
    (c) CNN
    (d) RNN

32. **Give a reason why increasing the Training Dataset causes Overfitting to happen at a later epoch in the training process?**

33. **If the mapping between the input and output in a dataset is not fixed but varies with time, then can this data still be modeled using a Neural Network? If so, how?**

34. **Recall that the parameter *p* in Dropout Regularization is the probability of retaining a node. What effect does decreasing p have on the Model Capacity, why?**

35. **Why does an un-balanced input sample cause problems in the training process?**

36. **What is the effect of Learning Rate on Model Capacity?**

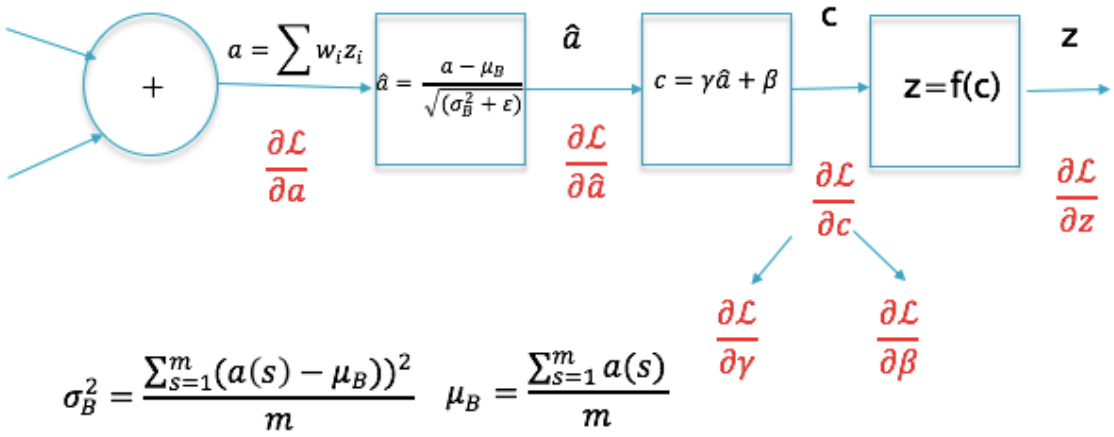
37. **On page 28 of Lecture 9, confirm that the probability of error for the example shown is indeed 0.026.**

38. **Explain how Batch Normalization also serves as a Regularization method.**

39. **The diagram below shows the forward computations for a node implementing the Batch Normalization algorithm. Assuming that the gradient flowing into the**

system from the right is $\frac{\partial \mathcal{L}}{\partial z}$ and its value is known, derive expressions for the following other gradients which are needed to do Backprop on the parameters (γ, β):
$\frac{\partial \mathcal{L}}{\partial \gamma}, \frac{\partial \mathcal{L}}{\partial \beta}$.

Hint: First compute $\frac{\partial \mathcal{L}}{\partial c}$.



CNNs

40. What are the advantages of a CNN over a fully connected DFN for image classification?

41. If you were to train a model to classify between Cats and Dogs, can you use the same model to classify between Roses and Tulips, explain. If no, can you still use a portion of the model?

42. Assume that you have dataset consisting of 3D color images (for example CAT scans belong to this class). How would you modify the CNN architecture that you learnt about in order to process these images? What would the Filter and Activation Maps look like?
Can you use the same design to process video segments?

43. Give three reasons why taking a small filter and convolving it over an image works better than using a single large filter than spans the entire image.

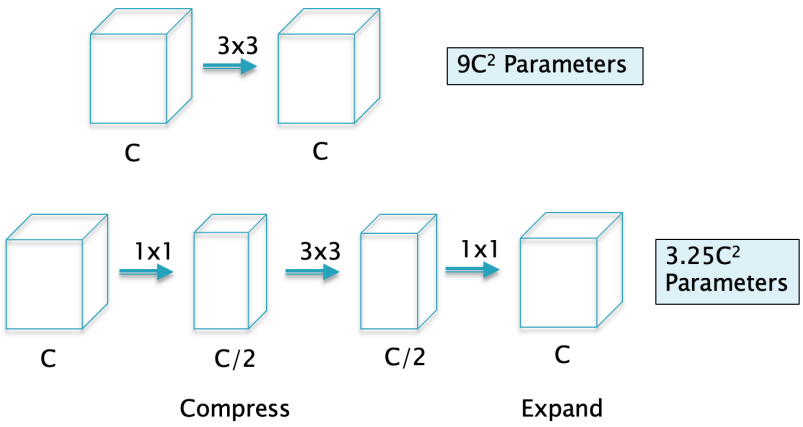
44. We increase the capacity of a Dense Feed Forward Network by adding layers or nodes, which increases its number of parameters. A CNN has higher capacity than an equivalent DFN, and yet has smaller number of parameters. Is

this a contradiction? Why do you think this is the case?

45. Consider a ConvNet composed of 3 convolutional layers, each with 3X3 Filters. The first layer has 100 Activation Maps, the middle layer has 200 and the last layer has 400 Activation Maps. The input images are RGB images of 200X300 pixels.
    a. What are the total number of parameters in the ConvNet?
    b. Assuming a stride of 1, compute the size of the Layer 1, 2 and 3 Activation Maps.
    c. If the Layer 2 and Layer 3 Activation Maps are constrained to be of the same size as that of Layer 1, what is the zero padding P required (assume stride =1)?

46. In Max Pooling or Global Max Pooling we are throwing away information by choosing a single Activation from a group. Yet they lead to improved performance, what is the reason for this? Are there cases where doing Max Pooling is not recommended?

47.



    a. The figure above shows the use of 1X1 filters in order to reduce the number of parameters in the model. Verify that the number of parameters reduces to $3.25C^2$ for this model (you may ignore the bias parameters for this calculation).

    b. Compare the number of computations required in both the figures to check whether smaller filters are also effective in making the models run faster.

48. Suppose I am using Data Augmentation to modify images before feeding them into my model. Would I use Fast Feature Extraction Method 1 or Method 2? Explain.

49. **A 1x1 CNN filter is said to be similar to a Dense Feed Forward stage. Can you explain why?**

50. **The best performing CNNs have a pyramidical shape (see Lecture 12, Page 23 for example), with increasing depth and decreasing cross-sections. How does this help?**

51. **What is the main innovation in ResNets as compared to older architectures? Explain why this innovation leads to better models.**

52. **Why are CNNs not the best architecture for finding patterns in sequence data?**

53. **The size of an Activation Map in a CNN can be reduced either by using 2x2 Max Pooling or by using regular convolutions with stride 2. State at least one benefit of using one of these techniques over the other.**

54. **When going from a convolution layer to a Feed Forward Layer, why is the Global Max Pooling operation preferable to the Flatten operation?**

55. **The convolution layers in a CNN contain information about the positions of detected objects within an image while this information is lost in the feed forward layers. Is this statement true or false? Explain.**

56. **What problem does Zero Padding in CNNs solve?**

57. **If we increase the size of an Activation Map in a CNN, does that lead to an increase in the number of parameters as well?**

58. **Give at least two reasons why CNNs need a smaller training dataset as compared to DFNs.**

59. **Why do CNNs take longer to train as a compared to a DFN?**

60. **Transfer Learning enables us to use a much smaller training dataset, why is this so?**

61. **How does Feature Extraction differ from Fine Tuning when doing Transfer Learning?**

62. **When is it advisable to do Feature Extraction using Method 2 as opposed to Method 1?**

63. **Why are smaller filter sizes preferred in modern CNN designs?**

64. Depthwise Separable Convolutions lead to a decrease in the number of computations by a factor of approximately $\frac{1}{D^2}$ where D is the filter size. Is this statement true or false?

65. The designers of ResNet noticed that the Training Loss was actually smaller for a 20 layer CNN as compared to a 56 layer CNN. Why did they consider this to be a surprising result? What did they conclude from this?

66. How do ResNets improve gradient propagation in a network?

67. In addition to better gradient flow, give two other reasons why Residual Connections make it easier to train a CNN.

RNN/LSTMs

68. A CNN can also be used process sequences, but it is not able to handle variable length sequences while a RNN can? What architectural feature in RNNS enables this capability? Can a RNN handle arbitrarily long sequences?

69. Come up with an architecture than can be used to process video segments by using a combination of RNNs and CNNs.

70. Why are RNNs subject to the Vanishing Gradient Problem? How was this addressed?

71. How is pattern recognition in RNNs different from that in CNNs?

72. Explain the function of the Input and Forget Gates in an LSTM, and why they are required (you can reference the figure on Page 23 of Lecture 14).

73. How is the gradient propagation in LSTMs similar to gradient propagation in a CNN with Residual Connections?

74. How do RNNs deal with variable length input sequences during the training process?

75. Why are the computations when training an RNN/LSTM done in a serial stage by stage fashion?

76. Why are bi-directional RNNs not appropriate to processing real time data sequences?

77. What problem occurs when we try to train a RNN on a very long sequence?

78. **What was the main innovation in LSTMs that enabled it to handle much longer sequences than a RNN?**

79. **What are the benefits of adding additional layers to a RNN/LSTM?**

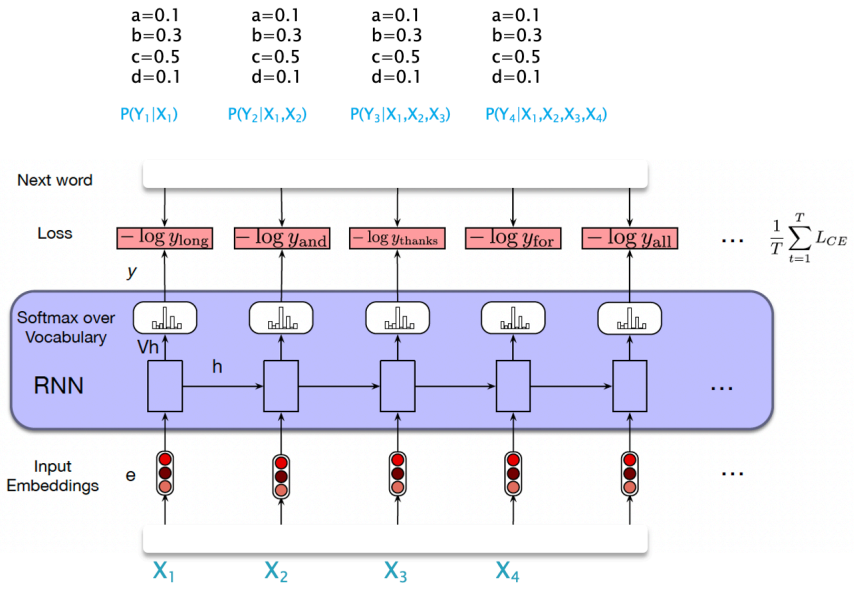
Natural Language Processing

80. **What is a Language Model, why is it useful?**

81. **Show that the following two definitions of Language Model are equivalent:**
a) Given a sequence of words $(w_1,...,w_N)$, a Language Model predicts the most probable next word w_{N+1} in the sequence.
b) Given a sequence of words $(w_1,...,w_N)$, a Language Model can be used to compute the probability $p(w_1,...,w_N)$ of that sequence occurring in the language

82. **Why is the architecture of a RNN/LSTM well suited for building a Language Model?**

83.



Consider a trained Language Model shown in the figure above with a vocabulary consisting of 4 words a, b, c, d. When subject to an input (X_1=d,X_2=b,X_3=a,X_4=d) it predicts the probability distributions for the next words, as shown in the figure.
(a) Using this information compute the probability of this sequence occurring (given that the first word in the sequence X_1=d).

84. **Explain the difference between the Decoder in an Encoder-Decoder model when operating in the Training mode as opposed to the inference mode.**

85. **Why doesn't Greedy Decoding work well for NLP? How does the Beam Method solve this problem?**

86. **What is the defining characteristic of an Auto-Regressive generation model?**

87. **When is it better to use pre-trained word embeddings, as opposed to finding the embeddings as part of the model training?**

88. **What is the benefits of using a RNN/LSTM model for doing Text Classification, as opposed to a Bag of Words based model with no recurrence?**

89. **Draw a block diagram of an Encoder Decoder system that can be used to answer questions about a given image. So the input consists of an image + question and the output is the answer to the question.**

90. **The Google Machine Translation system fed the input words into the Encoder part of the model in reverse order. Why did this lead to an improvement in translation performance?**

91. **Why was the Attention mechanism introduced into Machine Translation systems, and how did it help to improve performance?**

92. **A basic problem in building Neural Networks models that can do speech transcription from audio signals, is the huge mismatch between the number of elements in the input audio sequence (which can be in the thousands) and the number of elements in the output word sequence (which can be just a few words long). How was this problem solved?**

Transformers

93. **What were some of the problems with the RNN/LSTM architecture that the inventors of the Transformer were trying to address?**

94. **The Self Attention mechanism allows the representation of one of the words in a sequence, to be influenced by the other words in that sequence that come before it. Is this statement true or false?**

95. **Why does the Transformer Encoder incorporate a Dense Feed Forward module in addition to the Self Attention module?**

96. **Why are multiple Heads needed in the Self Attention module?**

97. **The number of parameters in a Transformer remain the same irrespective of the length of the input sequence. Is this statement true or false? Do the number of parameters increase if we add additional layers to the Transformer?**

98. **What are the number of parameters in a Transformer Encoder with the following design:**
    Word embedding dimension: 32
    Number of Attention Heads: 2
    Dense Dimension: 64
    Number of Layers: 10
    Sequence Length: 300

99. **Why are Positional Encodings needed in a Transformer? What are the benefits of a pre-computed Positional Embedding as opposed to a learnt Positional Embedding?**

100. **State at least two benefits of a Language Model using a Transformer vs that using a RNN/LSTM.**