Deep Learning

1. **What is deep learning**
2. **What is perceptron**
3. **What is ANN**
4. **Differnce between ANN vs Biological NN**
5. **Input layer, Output Layer**
6. **What is flatten layers**
7. **What is Dense Layers**
8. **What is Hyperparameters and Parameters**
9. **Activation function**
10. **Vanishing/exploding Gradient**
11. **Sigmoid function**
12. **Relu**
13. **Dying Relu**
14. **Leaky Relu(max(αz, z))**
15. **parametric leaky ReLU (PReLU)**
16. **Batch Normalization**
17. **Should we train large dnn from scratch?**
18. **Transfer learning**
19. **What is optimizer**
20. **Momentum optimization**
21. **Nesterov accelerated gradient**
22. **AdaGrad**
23. **RMSProp**
24. **Adam and nadam**
25. **Optimal Learning Rate [start with high learning rate and then reduce it]**
26.
27. **Softmax**
28. **What is RNN**
29. **What is LSTM**
30. **Avoid Overfitting through Regularization (l1 and l2 regularization)**
31. **Dropout, Monte Carlo Dropout**
32. **Max norm regularization**
33. **What is CNN [convulation+pooling+convulation+pooling+ fully connected]**
34. **Filter, feature map**
35. **What is Pooling layer, do it have weights?[Relu is used]**
36. **LeNet-5 (i C P c p c f f)(activation function tanh)**

37. **AlexNet (i c p c p ccc fff)(activatin function relu)**
38. **GoogleNet (have dropout layer)**
39. **VGGNet**
40. **ResNet (152 layers), resNet-34**
41. **SENet**
42. **Object Detection - YOLO**
43. **What is transformers**
44. **What is LLM**
45. **About t5, gpt-3, bert**

It is generally not a good idea to train a very large DNN from scratch: instead, you should always try to find an existing neural network that accomplishes a similar task to the one you are trying to tackle (we will discuss how to find them in Chapter 14), then just reuse the lower layers of this network: this is called transfer learning. It will not only speed up training considerably, but will also require much less training data