* ***Different types of Optimizers*** (At least 2 of them)  
   Compare them i.e. why A is better than B?  
   (a) **SGD** : Advantages and disadvantages of increasing the mini-batch size  
   **Q.** Show that SGD is an unbiased estimate of the Gradient Descent direction.  
   **Q.** What are the conditions of learning rate to ensure convergence?  
   (b) RMSProp  
   **Q.** Why do we introduce ‘r’? What are the advantages over SGD?  
   (c) Adam
* Universal Approximation Theorem  
   (a) What does it state?  
   (b) Non-Convexity of Neural Networks  
   (c) How does it relate to the SGD method of training neural networks?
* Write the ***Softmax*** function. Prove that it produces a probability distribution. Why is it used instead of *argmax*?
* Derive the cross-entropy error. How does it relate to log-likelihood?
* Comment on the computational cost and memory cost for training Neural Networks? (in terms of arithmetic operations)  
   (a) CC : Mini-batch size | # layers in the model | # hidden units per layer in the model  
   (b) MC : For forward and backward steps
* What are the advantages of using GPU over CPU for ML problems?
* What to do if the CPU is out of memory? What if GPU is out of memory?
* What are the different types of activation functions, hidden units?
* Advantages of 1st-order optimization over 2nd-order optimization?
* PyTorch v/s TensorFlow
* Automatic Differentiation
* Finite difference formula for computing gradients
* Linearization
* Xavier Initialization
* Compare and contrast BatchNorm and LayerNorm. Briefly state their mathematical equations and comment why they might work.
* ***Backpropagation Derivation*** - (Check for MSE caveat as well as the objective function)  
   (a) Multi-layer Fully Connected Network  
   (b) Multi-layer Fully Connected Network (with dropout)
* ***Basic PyTorch Commands*** (a) Take gradient with respect to a particular variable i.e. automatic differentiation  
   (b) Distributed Training (Synchronous and Asynchronous) - Algorithm and Code. Also, talk about the communication costs (if any!)  
   (c) If optimizer.zero\_grad() is not included, what happens?  
   (d) Command for moving the model from CPU to GPU, and vice-versa.  
   (e) Advantages of using float16/float32 in place of float64.
* ***Convolutional Neural Networks*** (a) Why are convolutions helpful? Particularly in the context of images?  
   (b) Describe different parameters i.e. padding, strides and state the mathematical construct with input/output channels.  
   (c) Explain 3D convolutions.  
   (d) Residual Networks (ResNet) from the HW-4
* Vanishing Gradient Problem  
   (a) What is the problem?  
   (b) What are the methods/ways to address it?
* Generative Adversarial Networks  
   (a) What is the objective function? Write the SGD for them.  
   (b) What is the Wasserstein GAN or the W-GAN algorithm?
* Recurrent Neural Networks (RNNs)  
   \*\*\*Computational Cost of these two methods i.e. how does it depend on total sequence T and the truncated length ***tau :*** (a) Backpropagation through Time (BPTT)  
   (b) Truncated Backpropagation through Time (TBPTT)  
   (c) GRU Equations  
   (d) LSTM Equations
* Different types of Regularization  
   (a) L1 penalty  
   (b) L2 penalty  
   (c) Dropout - Why is dropout (on test) ad-hoc? Talk about the two different methods of evaluation on test.  
   (d) ***Ensemble Models*** - Can their training be parallelized? If yes, what’ll be the efficiency of the process.  
   (e) Data Augmentation - different types  
   (f) Transfer Learning - example of ResNet done in HW4  
   (g) Early Stopping
* Difference between model parallelization and data parallelisation. When and why to use one instead of the other?
* What is efficiency in distributed training?
* Reinforcement Learning:  
   (a) Write down the Q-learning algorithm.  
   (b) How do we select actions i.e. explain the epsilon-greedy algorithm?  
   (c) If we take gradient with respect to *y* in the Q-learning algorithm, would we get an unbiased SGD algorithm? More generally, talk about biased v/s unbiased estimate of SGD.  
   (d) Policy Gradient algorithm.  
   (e) Actor - Critic algorithm. (Continuous and Discrete Control)  
   (f) What is Y? If Y(x, a) = Q(x, a), what will happen?  
   (g) Bellman Equation  
   (h) Deep Q-Learning  
   (i) Replay Buffer  
   (j) Double Q-Learning  
   (k) Tabular Q-Learning