# Answers to the theory questions.

## Q1. Explain the probabilistic interpretation of logistic regression.Why Does a model learn the probability distribution and not discrete values?

- if a model were to learn the discrete values (say 0 or 1),the loss function itself consist of discrete values only (unlike the case for probabilities,where there is a continuous loss fuction).This would cause the optimization problem to be intractable as finding gradients of a discrete valued function is hard(We'd have to use sub gradients,which is slower).
- allowing the model to output probabilities instead of discrete values allows usage of more sophisticated scoring metrics to be used,like the area under the ROC curve and such.
- logistic regression can be probabilistically modelled such that,out hypothesis $h_\theta(x)$ outputs the probability of the input $x$ belonging to class 1 (Let us assume).So, we have :

$$P(y = 1|x; \theta) = h_\theta(x) \;\; and,$$
$$P(y = 0|x; \theta) = 1 - h_\theta(x)$$

this can be compactly written as :

$$p(y|x; \theta) = (h_\theta(x))^y(1 - h_\theta(x))^{1-y}$$

hence,the likelihood of the parameters can be written down as :

$$L(\theta) = p(\vec{y}|x; \theta)$$
$$= \prod_{i=1}^{m} p(y^{(i)}|x^{(i)}; \theta)$$

this is the same as maximizing log likelihood,which is given by :

$$log(L(\theta)) = \sum_{i=1}^{m} y^{(i)} log h(x^{(i)}) + (1 - y^{(i)}) log(1 - h(x^{(i)}))$$

This maximization is done by gradient ascent. ## Q2.

momentum in SGD involves using the previous gradient in the present update as well.the previous gradient is included by weighting it with a hyperparameter $\beta$

hence the momentum can be summarized as follows :

$$w^{[l-1]} := w^{[l-1]} - \alpha V_t$$

where,

$$V_t = \beta V_{t-1} + grad_{new}$$

an intuition that can be given is that of a ball rolling down the bow shaped optimization curve, with $V_{t-1}$ denoting the current velocity and $grad_{new}$ denoting the accelaration of the ball.

## Q3.

**a) Why are mini-batches used?Also,compare mini-batch vs non-minibatch.**

Mini- batches are mainly used for the following reasons :

- taking the whole dataset to calculate loss and backpropagate using gradients can be very time consuming.Hence mini batches are used to reduce the cost of computing average loss/gradients.
- evaluating gradients for back prop using mini batches actually adds noise to the gradient step, and hence helps in not settling at a local optimum.
- choosing the size of the mini batch is extremely important.if we choose a very low batch size ,it will lead to oscillation of the weights around the minima(due to the fact that there is a lot of oise),thereby not settling at global minimum.if we choose a very large batch size,it causes the optimization step to be very slow.

**b) Why is symmetric weight-initialization problematic?**

- Symmetric weight initialization causes the weights to be updated in the exactly same fashion,which implies that neurons of the same layer will have the exact same weights,and hence there will be no variation in the outputs of each layer.(this would be equivalent to replacing all the neurons in a layer with a single equivalent neuron).

**c) explain regularization.How does it help in reducing over fitting?**

- regularization penalizes the model for having more weights or having weights with large norms. Hence it helps in simplifying models,because the number of weights that actually cause a difference in the output are minimal.hence,this reduces overfitting.
- Other ways to reduce over fitting are :
  - get more data.

- apply dropout to networks.
- use larger value of the decay constant.
- early stopping,although it gives a positive bias to validation accuracy.
- change the architecture of the model itself

**d) explain batch norm.**

batch norm is the process of normalizing the activations of every layer to a mean and a standard deviation.this helps to classify unseen data more accurately as this reduces internal co variate shift.this also helps in smoothening the optimization function,hence causing gradients to flow easily and cause faster optimization.we can also use higher learning rates due to this,thus causing faster convergence.Batch norm also has a regularizing effect(due to noise added while using mini batches).

## Q4.

**dimension of output is given by =** $\lfloor \frac{(N_1+1)}{2} \rfloor \times \lfloor \frac{(N_2+1)}{2} \rfloor$

**number of parameters :**

- number of weights in a $3 \times 3$ kernel $= 9$
- number of such kernels for a single output channel $= 3$ (Number of input channels)
- number of such 3D kernels : 8 (Number of output channels)
- Hence, total parameters $= 9 \times 3 \times 27 = 216$.