MLSP Assignment5,
Q1:
ADAGRAD
In normal gradient descent all parameters has same learning rate. Adagrad uses different learning rate for different parameters, at every timestep t.
Adagrad is for adapting the learning rate to the parameters. It performs larger updates for infrequent params, and smaller updates for frequent parameters.
It is well suited for dealing with **sparse data.**

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,ii} + \epsilon}} \cdot g_{t,i}$$

G_t here is a diagonal matrix where each diagonal element i, i is the sum of the squares of the gradients w.r.t. θi up to time step t[11], while is a smoothing term that avoids division by zero (usually on the order of 1e − 8). Without the square roots, algorithm performs worse.

Its main benefits is that we dont need to manually tune the learning rate.
Its main weaknes is sum in denominatopr keeps getting large, and hence actual update after some iteration will be almost zero because of increasing denominator.

b. **ADADELTA:**
Problem with adagrad is its monotonically decreasing learning rate. A bettter approach is to accumulate past gradients of some fixed window size w. For that we need to store all these w gradients, which is highly inefficient. So here we take exponential averaging of gradients, sum of gradients of previous steps, with some weight.
Denominator(t) = gamma * denomiantor(t-1) + gradient(t) ^ 2
gamma is set to be around 0.9.
Units in these updates do not match. So a correction term is introduced. another exponentially decaying average is defined, sum of squared parameter updates:

$$E[\Delta\theta^2]_t = \gamma E[\Delta\theta^2]_{t-1} + (1-\gamma)\Delta\theta_t^2$$

This is same as RMS(root mean square). Final update is:

$$\Delta\theta_t = -\frac{RMS[\Delta\theta]_{t-1}}{RMS[g]_t} g_t$$
$$\theta_{t+1} = \theta_t + \Delta\theta_t$$

c. RMSPROP:
Intution behind Rmsprop and adadelta are same. They were developed independently by two peoples.
Denominator(t) = gamma * denomiantor(t-1) + (1-gamma) * gradient(t) ^ 2. Here the rms(update) term is not used in numerator.

D. ADAM:

It is a mix of **momentum** method and **rmsprop** method.

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$
$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

M is exponential averaging of momentum. V is exponential averaging of denomiantor term, square of gradient at every time step.
M and v are initilised with zero vector. Because of zero initialization, these are initially biased towards zero. To decreae these biases, computing of first and second moment is done as:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}$$

Final update is:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t$$