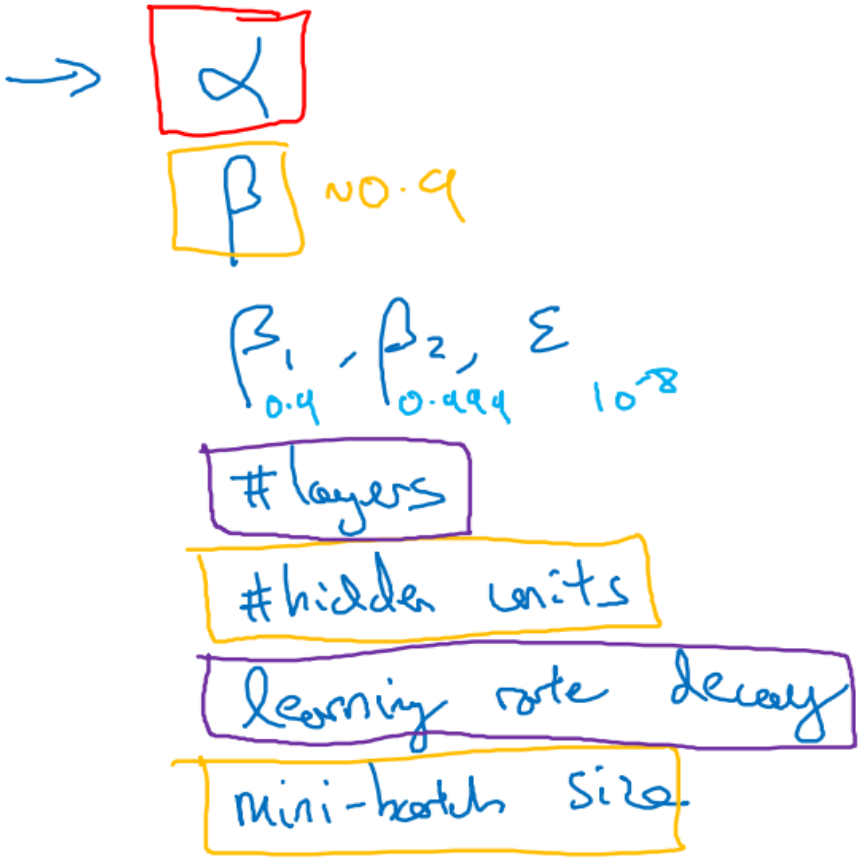
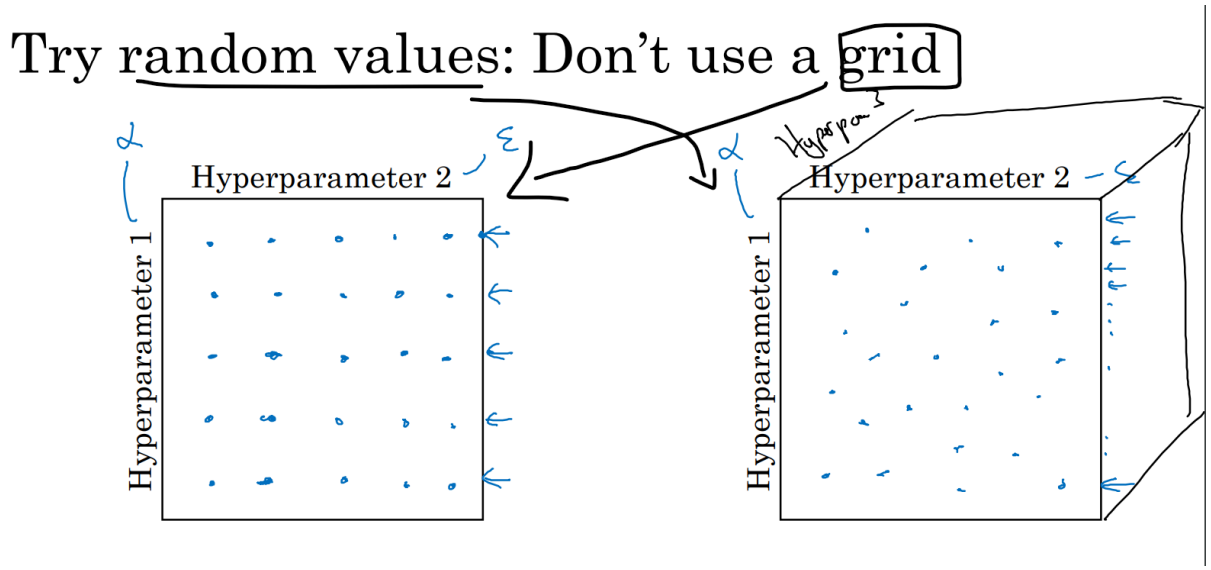
W3 C2

hyper parameters



Order of significance - red -> yellow -> purple
Uncircled is Adam Optimisation with default hyperparameters

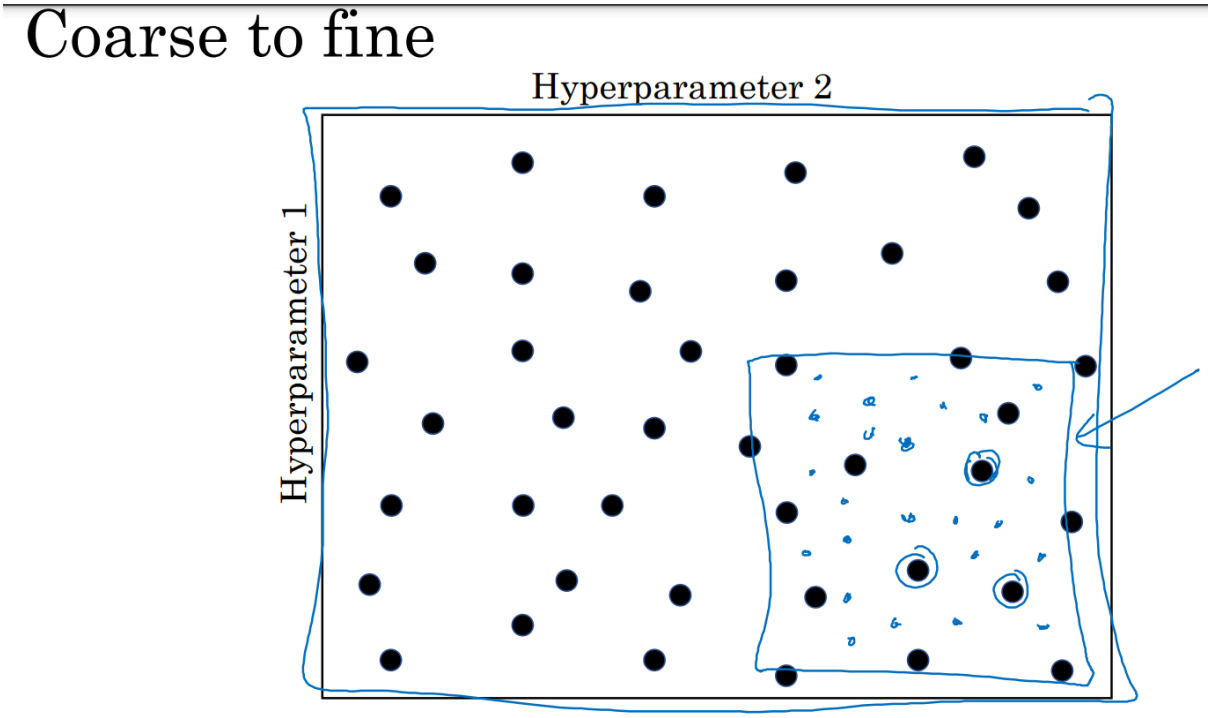


Grid allows to take a specific number of hyperParameters

here, 5 of each alpha and epsilon are taken, Important parameters can't be identify as same no. of pts of each hyperparameter is taken,

Randoms finds hyperparameters which will be the most important for your specific problem as it can explore a wider range of values for each hyperparameter also, it can increase the likelihood of finding a value that works well.Random can lead to missing out on some important points

Best ways coarse

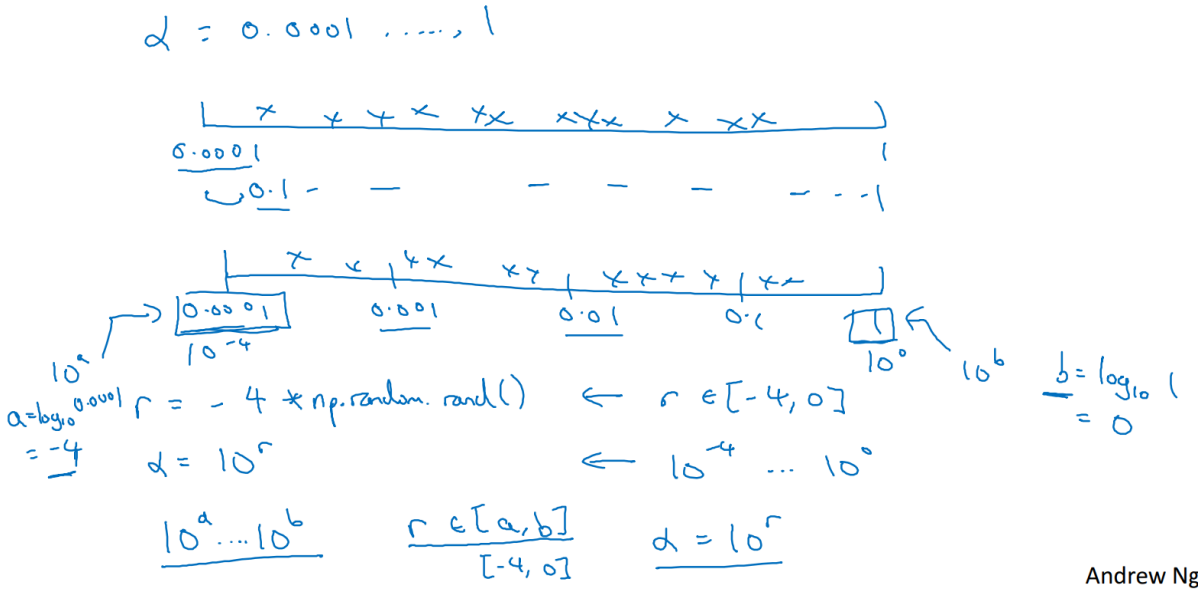


The Area with points having better performance than others And the same random values is performed on this area

Random vals can be highly non uniform,
Eg for alpha : 0.0001 to 1, vals bw 0.0001 and 0.1 might be less than 0.1 to 1
The scale for random values is the log scale
It helps to set random values to log values so that there is some uniformity

As per this scale Random val of hyperparameter = 10^r;
Formula R and range for r is given,

The hyperparameter range depends on range of R

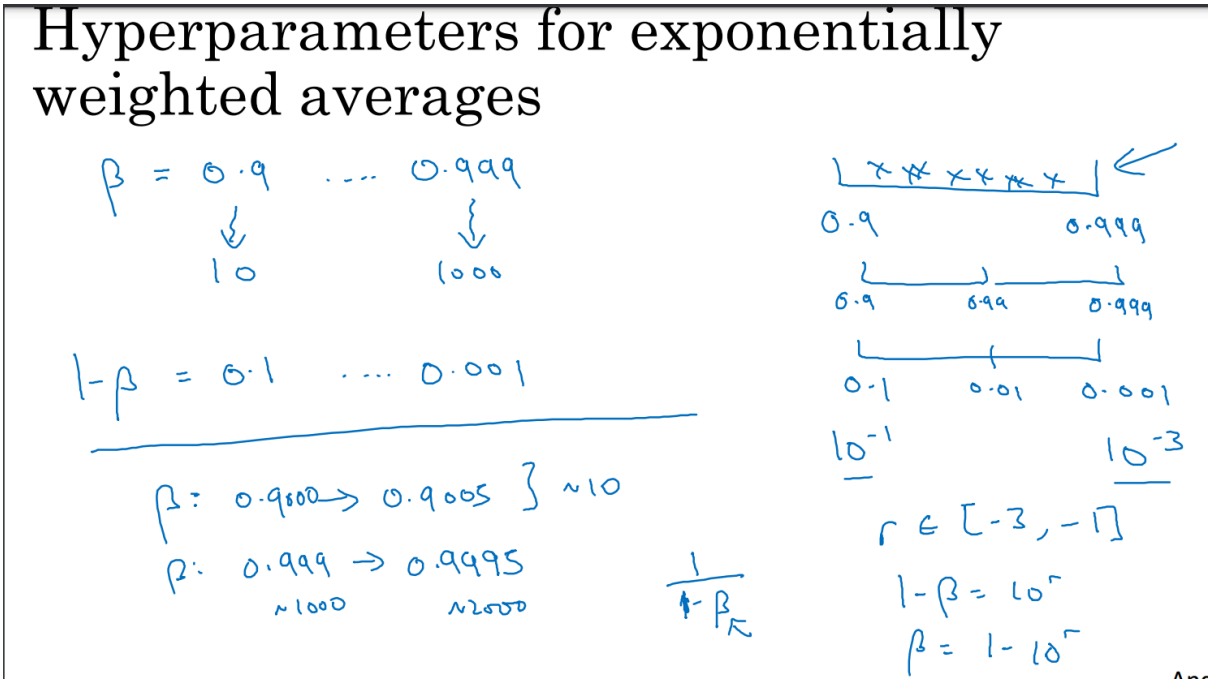


Andrew Ng

The log scale with values ranging from 0.0001 to 1 Response to 10^a to 10^b

To get a and b, 10^a = 0.0001and rest is shown in the image

we define a range of values that can be represented by logarithms rather than every possible log value between 0.0001 and 1 and sample uniformly at random within that range.



Suppose we suspect beta to be in the range 0.9 to 0.999,

then we can say that 0.9 If they average over last 10 days were as 0.999 is average over last 1000 days
The scale of beta needs to be Similar to the 1st sample scale,i.e., more like 0.1 etc.

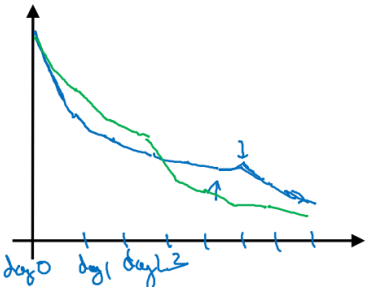
1-beta is Introduced for this reason,as 1 - beta approaches 0, Values need to be Carefully observed

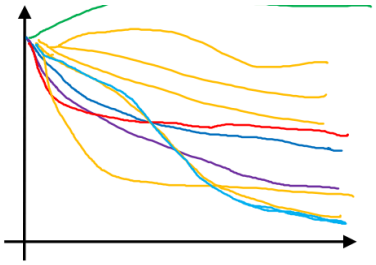
it is important to distribute the samples more densely in the region where beta is close to 1
small change in beta can have a significant impact on the results of the algorithm- as Given in the image 0.9000 to 0.9005 change holds no importance relative to 0.999 to 0.9995

Beta and ‘r’ relation are given

Hyper parameters need Occasional revaluation / Retesting such as in NLP, Vision, Speech, Ads, logistics, etc.
- Intuitions do get stale. Re-evaluate occasionally.

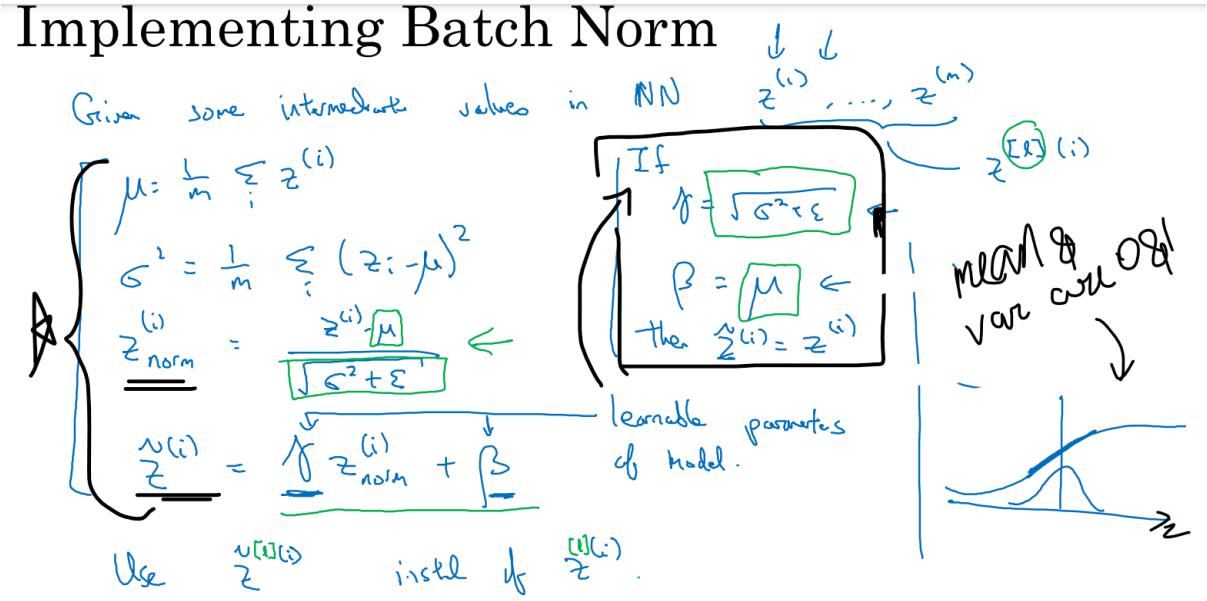
Types of tuning [Tuning and revaluation/ Retesting ain’t same]
- Pandas approach - Babysitting one model - computational resources are less
- Caviar approach - Training many models in parallel that run by itself





Caviar to right, pandas to left

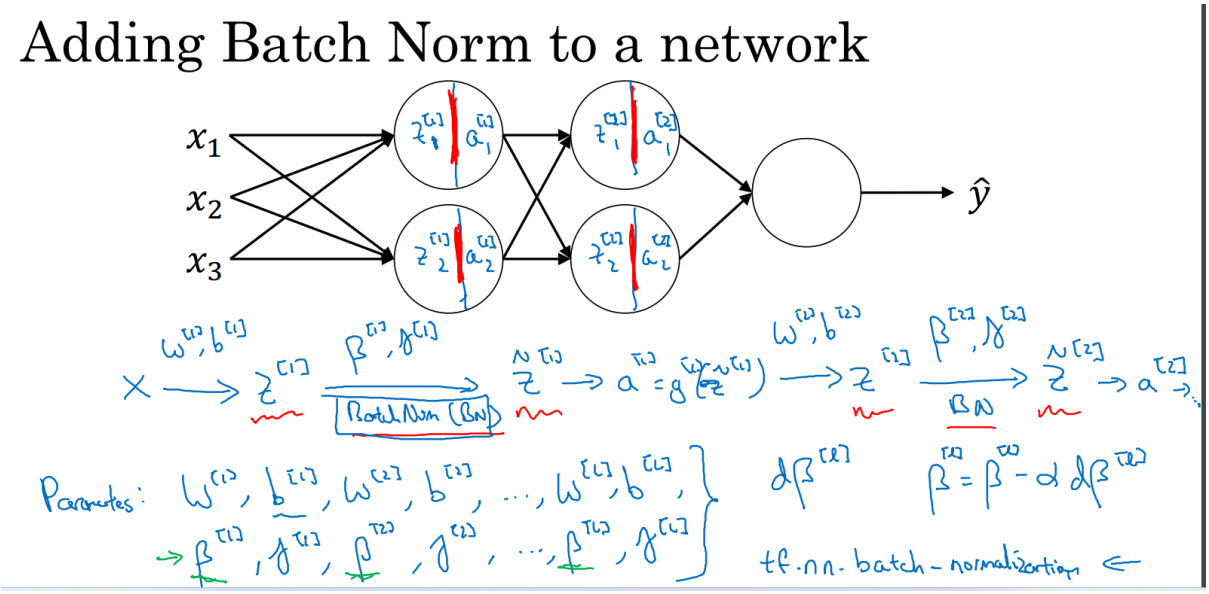
hyper parameter search [Find optimal values of hyperparameters] is easier by normalisation, Normalizing input values [activations] for the hidden layers makes Training w and b of this layer efficient



batch norm - normalising activations for which ‘norm Z’ isnt enough As it guess mean and variance of 0 and 1 which gives a graph as shown in the image

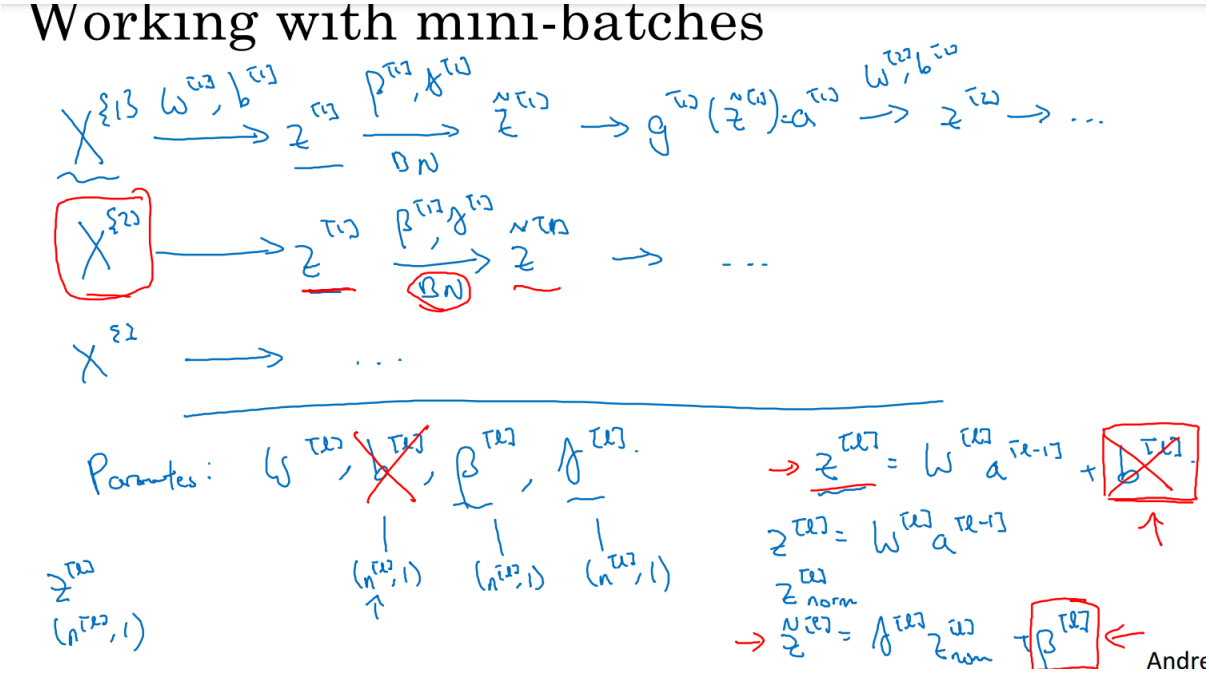
Z tilde is the solution - Gives mean and variance values different than 0 and 1 also, It contains learnable parameters of the model

if stt in the image is fulfilled, then, z tilde equals z norm



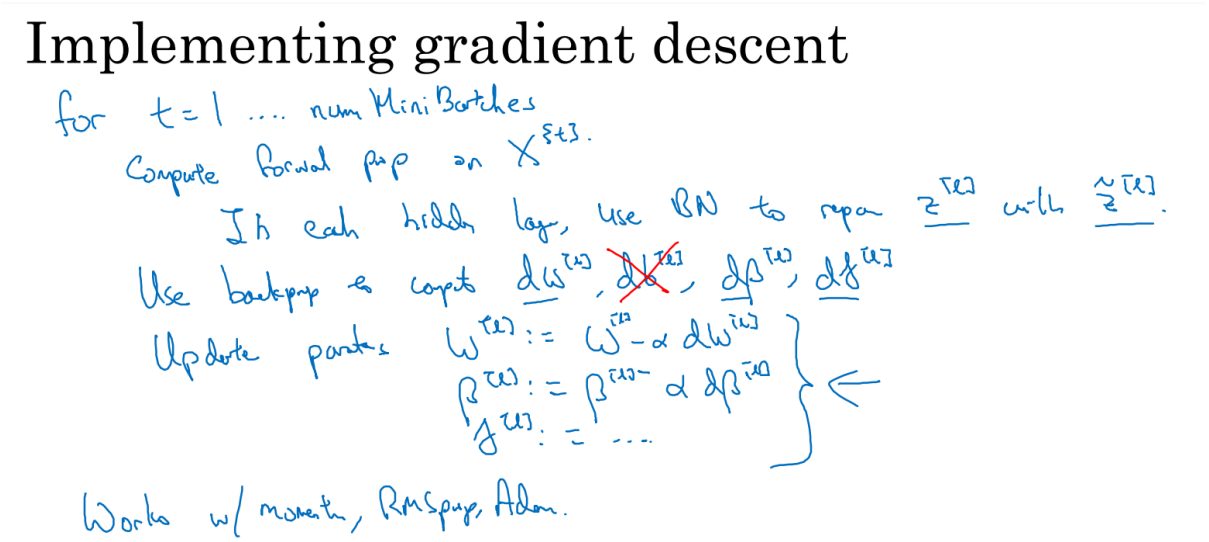
As the network shows that instead of using z we can use

Normalised version - z tilde Which calculates activations
Z and z tilde form single layer
This much only to learn from the above image



Batch norm deals with mini batches
as seen how the network differs for the different Mini batches

Note - b which is a constant doesn't contribute significantly and can be neglected

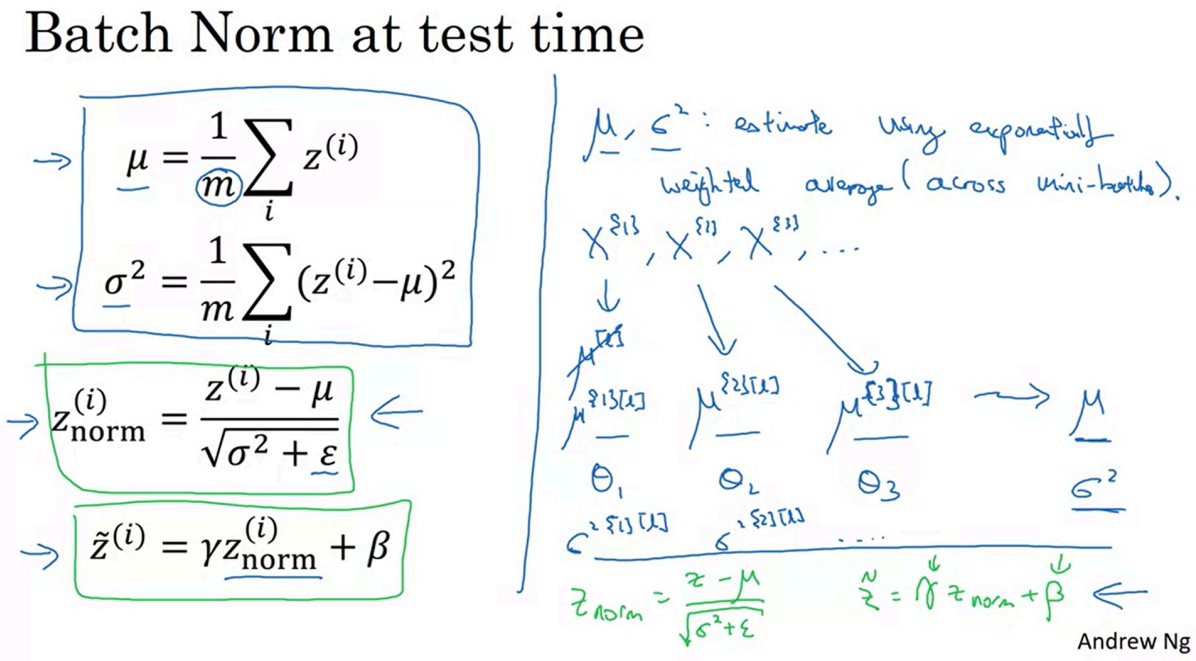


In each hidden layer those BN to compute Z tilde from Z

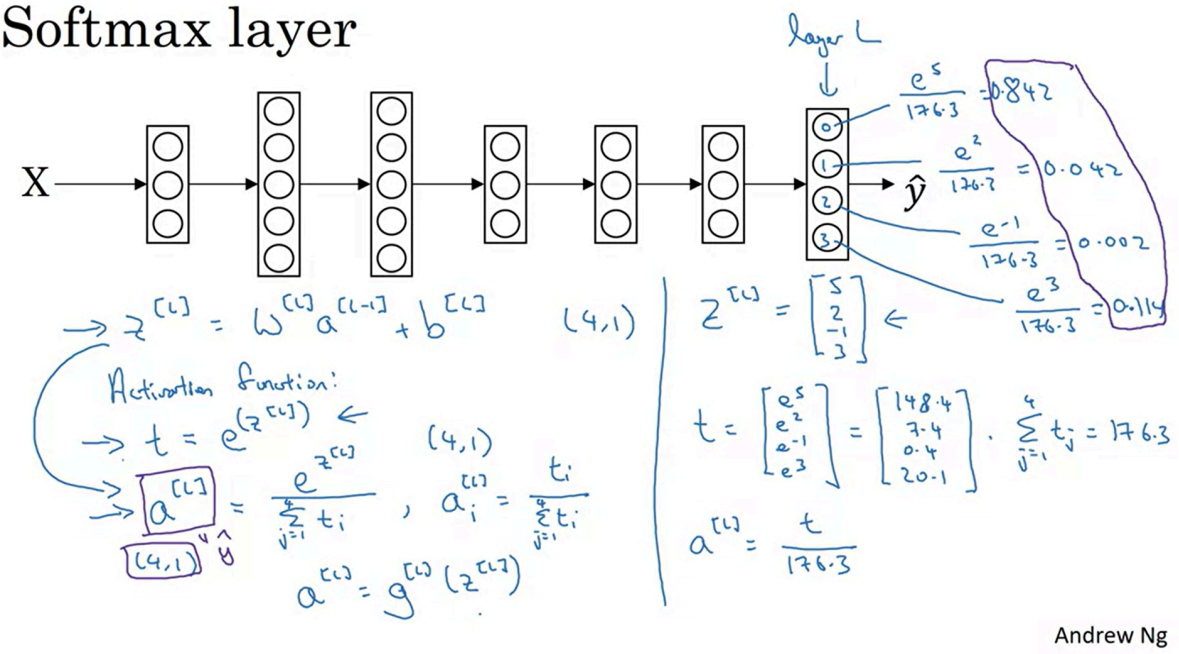
covariate shift- when distribution of input data of training data and the test data have different distributions. Causes problem to the network to generalize well to new data if the distribution has changed.

Batch normalization is solution - reduces the amount that the distribution of hidden unit values shifts around - by standardised mean and variance.

It also has a slight regularization effect by adding noise to the hidden layers' activations - noise in Z vals due to Mini matches have diff w and b as these Get updated often
This noise leads to the output hidden unit not being dependent on anyone input hidden unit - Like regularisation
 Mini batch size and regularisation are inversely proportional cus noise decreases



The mini batch size won't be the test size as well
To Implement batch norm at test time we keep a track of mean and sigma squared during train time via running avg/ exponentially weighted average



For binary classification only required 2 classes

for multi class such as cats dogs and baby chick classification,
They output is k neurons ; k=no. of classes
Sum of the neuron values is 1

For computition we first introduce Z[L] then, A temporary variable ‘t’ for activations
shape of a, t are same as Z, here, (4,1)

the final formula a a[l]=g(z[l]) given to Summarise all the computation

Softmax Logistic regression is generalization Of Logistic regression to more than two classes ,Some softmax egs are illustrated on the next pg

Refer image below
Forward propagation - zL = WL * aL-1 + bL
Backward propagation - dzL = y_hat - y
dzL and zL have same shape

The Loss function given is according to the column Matrix above it
To Minimise loss function we need big y_hat.
The Colum vector for this loss had probability of cat as 1

Y matrix which contains all such Colum vectors whose probability vary can be cat, dog or none .shape of Y is (4,m)