1.

a) The momentum parameter ends up being a rolling average of the previous gradients. How many, depends on the value of β_1. This way the gradient should not jump in wrong directions constantly, but rather maintains the average direction of the past 10 or so batches allowing for a more direct route to the optimal result.

If the rolling average of v is low then that means that gradients are very low, which means that learning is not happening. So when we divide by v we can increase learning rate in a plateau to hopefu get out of it quickly. Conversely, if the gradients are large, vis large and our learning rate is reduce to not overshoot our target

b) Since we disable a proportion of p_{drop} units, we want to scale the the remaining nodes in such a way that the total value remains about the same as if the nodes had not been disabled at all, i.e. the factor is $(1 - p_{drop})$.

Dropout is applied so that the network can't rely on single neurons to make a single prediction and as such there are an ensemble of neurons that can do the prediction. At test time this does not make sense as we are no longer training but rather want the best and predictable results so dropout should no longer be active as it is mainly used as a means of avoiding overfitting.