**DL theory : Assingments-12**

1. Unsqueeze is a function in PyTorch that is used to add an additional dimension to a tensor. This can help to solve certain broadcasting problems by allowing for tensors of different dimensions to be broadcasted together.
2. Indexing can be used to achieve the same result as unsqueeze by adding a new axis to the tensor and then indexing into that new axis.
3. To show the actual contents of the memory used for a tensor, you can use the .data attribute of the tensor in PyTorch.
4. When adding a vector of size 3 to a matrix of size 3x3, the elements of the vector are broadcasted across each row of the matrix.
5. Broadcasting and expand\_as can result in increased memory use because it requires creating a new tensor with the expanded dimensions.
6. One way to implement matmul using Einstein summation notation is to use the einsum function with the equation "ij,kj->ik", which represents the sum of the product of elements i,j of the first tensor and elements k,j of the second tensor, resulting in a new tensor with dimensions i,k.
7. A repeated index letter on the lefthand side of einsum represents a summation over that index.
8. The three rules of Einstein summation notation are: (1) an index letter appearing once in a term represents a single array element, (2) an index letter appearing twice in a term represents a summation over that index, and (3) an index letter appearing in both terms represents a contraction over that index. These rules allow for a compact notation for expressing complex tensor operations.
9. The forward pass in a neural network is the computation of the output of the network given an input. The backward pass is the computation of the gradients of the output with respect to the parameters of the network, which are used in the optimization process.
10. We need to store some of the activations calculated for intermediate layers in the forward pass because they are needed for the backward pass to compute gradients.
11. A downside of having activations with a standard deviation too far away from 1 is that it can cause the gradients to become either too large or too small, making the optimization process unstable.
12. Weight initialization techniques such as Glorot or He initialization can help avoid this problem by initializing the weights with a standard deviation that is close to 1, which can help to keep the activations within a stable range.