[Q1] But we dont use it here why? Why is it an array of 10 elements?

#Ans - We dont use label=int(f) because we are using one hot encoding where categorical values are shown as binary values shown in an array of 10 elements and it is necessary for cross entropy loss function. It is an array of 10 elements because there are ten folders named (0-9).

[Q2] Why convert to numpy array?

# Ans- Numpy arrays are easy to deal with compared to lists

[Q3] Why are we doing this and what does this type of slicing result in?

# Ans -In numpy arrays , we can use : to get specific values in array . Example - 0:3 gives values at 0,1,2 index (exculding 3)

# so to get all values we use (0: or :) so we use : for image , height , width . For channel we use 0 because we want to convert 3 channel image( RGB ) to single channel image in grayscale

[Q4] Why are we reshaping the data?

# Ans - X.shape[0] is the number of images and X.shape[1]\*X.shape[2] is the number of pixels (length\*breadth) in each image .

# We are converting 2D image to 1D array because it is easy to process array in 1D

[Q5] What is the learning rate?

#Ans-Learning rate shows how the weights are updated during epochs . If learning rate is high , weights are updated in large steps and if learning rate is low , weights are updated in small steps.Basically size of steps taken to reach minima in the loss function.

[Q6] Why are the dimensions of the weights and biases the way they are?

#Ans- Weights are basically the strentgth of connection of a neuron to another neuron in the previous layer .

#Baises are value added to the weighted sum of a neuron , the weighted sum will be equal to b when input is 0.

#The dimension of weights is 2D as it is differnet for each combination of neurons in previous and current layer

#The dimension of biases is 1D as it is unique for each neuron in the current layer, not depend on previous layer

[Q7] What is broadcasting and why do we need to broadcast the bias?

#Ans- We can use broadcasting in NumPy to perform operations on arrays of different shapes , NumPy stretches the smaller array over the larger one so make them compatible.

#To calculate the weighted sum of a neuron , we need to add both weights and baises , which have differnet dimensions , hence broadcasting is needed.

[Q8] What is np.random.randn? What's the shape of this matrix?

Ans- Creates a 2D array of random numbers drawn from a standard normal distribution (mean 0, std 1)

Its 2D as it is dependent on both previous and current layer neurons(hidden and input layer)

The np.sqrt(2/input\_neurons) is called He initialization, it prevents the ativation from becoming too small or too large.

[Q9] What are activation functions and why do we need them?

Ans- Activation functions are necessary to convert the weighted sum z of a neuron to an output value a of the neuron.

the weighted sum has a value of z=i∑​wi​xi​+b (where wi are the weights, xi are the inputs, and b is the bias).

But the output is not in the range as weighted sum so we use activation function to convert it to a value in a specific range.

[Q10] What is the softmax function and why do we need it? Read up on it

Ans - Softmax is also an activation function which is primarily used in the output layer of a neural network.

It converts raw weighted sums to probabilities by taking the exponent of each weighted sum and dividing it by the sum of all the exponents.

It gives an output between 0 and 1 , and the sum of all outputs is equal to 1.

[Q11] What are loss functions and why do we need them?

Loss functions show how wrong our model is in its predictions , they use predicted and actual values as input

and gives output showing how wrong the prediction is . To make the model better , we need to minimize the loss function

which means we take its derivative and keep subtracting it from loss function to reach minima.

[Q12] What does the output choice we have here mean? It's an array of 10 elements per image, but why?

Ans - The output choices are 10 different types of fashion items (0-9), so the output layer has 10 neurons.

[Q13] Why are we subtracting the mean of the inputs?

Ans- By subtracting the mean from inputs , we are making the data centered around 0, this prevents gradients

from becoming too small or too large , which can lead to problems later

This is called mean normalization .Make model efficient

[Q14] Why are we using the softmax function here?

#Ans- As stated earlier , softmax is an activation function which is used to bring the weighted sum (output\_inputs) to a range of 0-1

[Q15] Why are we doing a forward pass here instead of just using the outputs from the forward function?

Ans - We do a forward pass inside backdrop because we need the most recent outputs , weights and biases to calculate loss and update weights

bringing from the forward function will give old values.

[Q16] What is the validation dataset and what do we mean by generalization?

Ans - Validation dataset is a part of the original dataset which is not used in model training , rather

it is used to check how the model does with unseen data. Generalisation means how well the model can perform

on unseen data like the validation dataset

[Q17] What are the parameters in the model and what do they mean?

# Ans - 1.Input\_neurons= Number of nurons in input layer (784) from the 28\*28 image

# 2.Hidden\_neurons= Number of neurons in hidden layer , this parameter can be changed to get better results

# 3.Output\_neurons= Number of neurons in output layer (10) for the 10 fashion items

# 4.Learning\_rate= The shows the rate at which weights are changed during training , this parameter can be changed to get better results

# 5.Epochs= Number of times the model trains on the dataset , this parameter can be changed to get better results

[Q18] Why are we using argmax here? Why is this output different from the output of the model?

# Ans - Argmax returns the index of the position which has the maximum value in an array.The axis=1 gives index of maxima in each row.

# The output of the model gives probabilities of each class (0-9) obtained from softmax function , this final output shows the class with maximum probability.