

Problem Set 2

Sharvita Paithankar

Question 1:

(a)

Parameters are the values that are learned by a model while training. The values get updated as the model processes the training data and it helps with making better predictions over time.

HyperParameters are parameters that are set before the training even begins. They can control how the model learns and it adjusts the learning rate as well as decide how many layers a neural network should have to train.

Parameters are updated based on training split to help the model improve the predictions. Hyperparameters are tweaked by the person after evaluating the model's performance on the validation split. It helps with adjusting the learning rate and figuring out a good batch size to get better results.

(b)

Item	P or HP?	Justification
A weight matrix w	P	It is learned during training
The learning rate	HP	Set before training, controls how much weights updates during training
A bias term b	P	It is learned during training
The minibatch size	HP	Set before training, decides samples per update
A non-linear activation function	HP	It is chosen before training, defines model architecture
The optimizer	HP	Set before training, determines how parameters are updated

(c) Two addition Hyper parameters:

1. The quantity of layers, or the number of neurons in each layer

- This establishes a neural network's general structure. The model can learn more complex patterns if it has more layers or neurons.

- It is selected before the training and stays constant.

2. Dropout Rate: To avoid overfitting, a portion of neurons are randomly removed during training as part of this regularization strategy.

- The dropout rate is set before training and reduces dependency on specific neurons.

Both of them are hyper-parameters since they have an impact on the model's learning process but cannot be changed during training.

Question 2:

(a) Observation: The validation loss goes down at first but then starts going up, while the training loss keeps getting smaller.

This means the model is overfitting. It is learning too much from the training data and not doing well on new data.

How to fix it:

1. Use Early Stopping – Stop training when validation loss starts increasing.
2. Add More Training Data – Get more data or use techniques to create new training examples.
3. Add Dropout Layers – These help the model rely on all neurons instead of just a few.
4. Make the Model Simpler – A complex model may memorize the training data instead of learning patterns.

(b) Observation: Validation accuracy improves very slowly, and the training loss never becomes very small.

This means the model is underfitting, which means it is too simple and not learning enough from the data.

How to Fix It:

1. Make the Model Bigger – Add more layers so it can learn more patterns.
2. Restart Training – Sometimes, starting over helps if the model gets stuck.
3. Train for More Time – Let the model learn for a longer period.
4. Reduce Dropout – If dropout is used too much, it might be stopping the model from learning well.

Question 3: