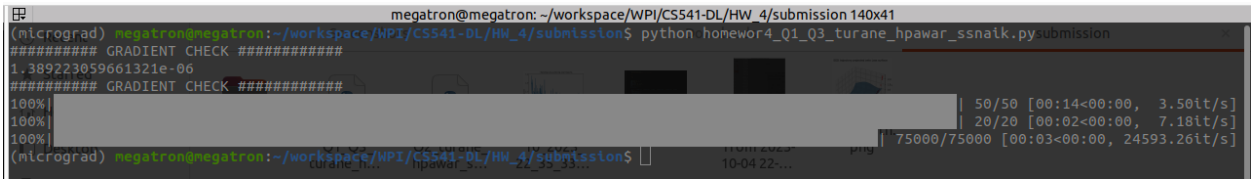
Homework_4 Answers

Tejas Rane: 901015176 | Hrishikesh Pawar: 901014710 | Shounak Naik: 728556739

Q1. **Feed-forward neural network**

a. Code file attached. Refer to the attachment homework4_Q1_Q3_turane_hpawar_ssnaik.py

b. Check-grad output:



Q2. **Feed-forward neural network with deep learning**

a. Code file attached. Refer to the attachment homework4_Q2_turane_hpawar_ssnaik.py

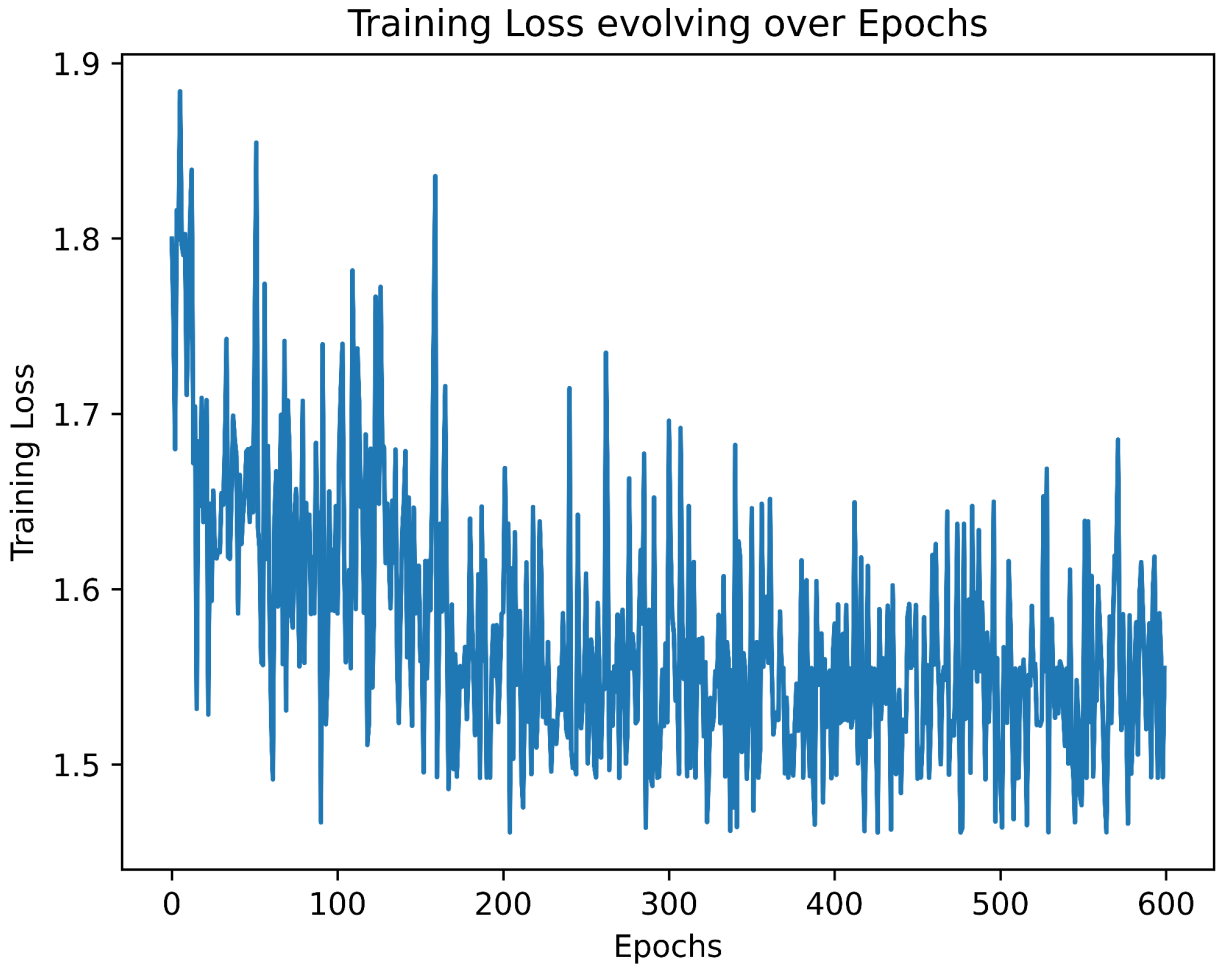
b. Please refer to the code file attached: homework4_Q2_turane_hpawar_ssnaik.py

```python
def findBestHyperparameters(self):
    """ Finds the best hyperparameters for the neural network.

    Args:
        None

    Returns:
        None
    """

    # Hyperparameters
    hidden_sizes = [64, 128, 256]
    hidden_layers = [3, 4, 5]
    epochs = [1000]
    minibatch_sizes = [32, 64, 128]
    learning_rates = [1e-4, 1e-3, 1e-2]
    alphas = [1e-3, 1e-2, 1e-1]

    best_val_loss = float('inf')
    best_model = None

    _total_runs = len(hidden_sizes) * len(hidden_layers) * len(epochs) * len(minibatch_sizes) * len(learning_rates) * len(alphas)
    _run = 0
    if epochs[0] >= 100:
        _epoch_sampling_rate = 100
    else:
        _epoch_sampling_rate = 5

    for hidden_size in hidden_sizes:
        for hidden_layer in hidden_layers:
            for epoch in epochs:
                for minibatch_size in minibatch_sizes:
                    for learning_rate in learning_rates:
                        for alpha in alphas:
                            _run += 1
                            print(f'Run {_run} of {_total_runs}')
                            model = Net(hidden_size, hidden_layer).to(DEVICE)
                            sgd = optim.SGD(model.parameters(), lr=learning_rate, weight_decay=alpha)
                            models = self.train(epoch, _epoch_sampling_rate, minibatch_size, model, sgd)
                            for i, model in enumerate(models):
                                val_loss, val_acc = self.validate(model)
                                if val_loss < best_val_loss:
                                    best_val_loss = val_loss
                                    best_model = model
                                    best_hyperparameters = [hidden_size, hidden_layer, (i+1)*_epoch_sampling_rate, minibatch_size, learning_rate, alpha]
                                    self.utils._print_hyp(best_hyperparameters)
                                    self.utils._logger(best_hyperparameters, best_val_loss, val_acc, best=True)
                                else:
                                    _hyp = [hidden_size, hidden_layer, (i+1)*_epoch_sampling_rate, minibatch_size, learning_rate, alpha]
                                    self.utils._logger(_hyp, val_loss, val_acc)

    print('##### Training completed! #####')
    self.utils._print_hyp(best_hyperparameters)
    best_hidden_size, best_hidden_layer, best_epoch, best_minibatch_size, best_learning_rate, best_alpha = best_hyperparameters
    model = Net(best_hidden_size, best_hidden_layer).to(DEVICE)
    sgd = optim.SGD(model.parameters(), lr=best_learning_rate, weight_decay=best_alpha)
    model = self.train(best_epoch, -1, best_minibatch_size, model, sgd, test=True)
    self.test(model[-1])
```

c. Screenshot displaying
   - Training loss evolving over the last 20 epochs of SGD.
   - Accuracy over test-set **87.889%**

```
Epoch 569, Train loss 1.61910130828619
Epoch 570, Train loss 1.618777807801962
Epoch 571, Train loss 1.6852410398423672
Epoch 572, Train loss 1.5545179843902588
Epoch 573, Train loss 1.5196732394397259
Epoch 574, Train loss 1.5856818817555904
Epoch 575, Train loss 1.523657288402319
Epoch 576, Train loss 1.5277439653873444
Epoch 577, Train loss 1.4662604182958603
Epoch 578, Train loss 1.5850142613053322
Epoch 579, Train loss 1.4947575442492962
Epoch 580, Train loss 1.5244834758341312
Epoch 581, Train loss 1.5542350299656391
Epoch 582, Train loss 1.5810888670384884
Epoch 583, Train loss 1.5056268312036991
Epoch 584, Train loss 1.5974675193428993
Epoch 585, Train loss 1.6153546459972858
Epoch 586, Train loss 1.5862427093088627
Epoch 587, Train loss 1.5564769469201565
Epoch 588, Train loss 1.5201383344829082
Epoch 589, Train loss 1.5588763430714607
Epoch 590, Train loss 1.5805335715413094
Epoch 591, Train loss 1.492688488215208
Epoch 592, Train loss 1.5992115810513496
Epoch 593, Train loss 1.6185808666050434
Epoch 594, Train loss 1.5476082004606724
Epoch 595, Train loss 1.4924100935459137
Epoch 596, Train loss 1.586248703300953
Epoch 597, Train loss 1.5592529475688934
Epoch 598, Train loss 1.4927169531583786
Epoch 599, Train loss 1.5549515187740326
Test loss: 1.5817795675873758
Test Accuracy: 87.88999938964844%
(microgrd) megatron@megatron:~/workspace/WPI/CS541-DL/HW_4
```

```
##### Training completed! #####
################################################
Best hyperparameters are:
Hidden size: 64
Hidden layers: 4
Epochs: 600
Minibatch size: 32
Learning rate: 0.01
Alpha: 0.001
################################################
Epoch 0, Train loss 1.7999575175344944
Epoch 1, Train loss 1.7539095245301723
Epoch 2, Train loss 1.6798102855682373
Epoch 3, Train loss 1.8161344341933727
```



Q3. **Mountains and valleys**

SGD trajectory projected onto Loss surface