CS 497 Midterm (Due by 10:00am on October 30)

Implementation (100 points)

Implement a neural network with one hidden layer and a softmax output layer, using PyTorch with the torch.nn and torch.optim modules. Evaluate the implementations on the same two 2D non-linear classification tasks: flower and spiral. Starter code and functions for generating the datasets are available at midterm.zip. The provided code also displays and saves images of the datasets and the trained model's decision boundaries. Make sure that you organize your code in folders as shown in the table below. Write code only in the Python files indicated in bold.

```
Midterm/
   nn1LayerExercise.py
   utils.py
```

PyTorch.NN Implementation

Coding effort: my implementation has 15 lines of code in nn1LayerExercise.py.

You will need to write code for the following:

1. **Model, Loss, Optimizer:** Define the model to be a NN with one hidden ReLU layer and linear outputs. You may want to use the torch.nn.Sequential container. Define the loss function to compute the cross-entropy – see loss functions defined in the pytorch.nn module. Define the optimizer to run SGD with the specified learning rate and weight decay.

2. **Gradient descent loop:** Write code for running gradient descent for num epochs. At each epoch, you will compute the model predictions using the model above, compute the loss between predictions and true labels using the loss function above, print the loss every 1000 epochs, zero de gradients through the optimizer object, then run backpropagation on the loss object.

3. **Hints:** Look at the "PyTorch: optim" section of the "LEARNING PYTORCH WITH EXAMPLES" tutorial https://pytorch.org/tutorials/beginner/pytorch_with_examples.html

Submission

Electronically submit on Cougar Course a lastname.zip file that contains the folder with two files nn1LayerExercise.py and utils.py.

On a Linux system, creating the archive can be done using the command:

> zip -r lastname.zip midterm.

Please observe the following when handing in homework:

---

1. Structure, indent, and format your code well.
2. Use adequate comments, both block and in-line to document your code.
3. Make sure your code runs correctly when used in the directory structure shown above.