STAT 453 **Name:** ____________________

SS 2025

Midterm Exam

03/06/2025

Time: 2:30 – 4:45 pm am (75 mins) **Email:** ____________________ @wisc.edu

Instructor: Yiqiao Zhong

Teaching Assistant: Zhexuan Liu

Access Code: ??????-ZHONG

This exam contains 7 pages (including this cover page) and 9 questions.
Total of points is 100.

By submitting this exam, I (the student)

- acknowledge that I am required to follow the academic integrity and conduct policies of UW-Madison.

Grade Table (for teacher use only)

Question	Points	Score
1	10	
2	10	
3	10	
4	10	
5	10	
6	10	
7	10	
8	10	
9	20	
Total:	100	

1. (10 points) Which one of the statements is NOT correct about the perceptron learning algorithm?

   A. Perceptron algorithm is an iterative algorithm.

   B. In the training loop, the error of each training example takes value $-1, 0$, or 1.

   C. The perceptron algorithm is not guaranteed to converge even for linearly separable data.

   D. The perceptron algorithm has a history of over 50 years.

   ____

   Solution:

2. (10 points) What is NOT a feature of PyTorch?

   A. PyTorch and NumPy share many similarities.

   B. PyTorch has built-in support for automatic differentiation.

   C. During the forward pass, a dynamic computation graph is created so that the model can implement back-propagation.

   D. When using PyTorch, we still need to manually work out the backward computation in a network definition.

   ____

   Solution:

3. (10 points) Which one of the statements is NOT correct about the multi-category cross-entropy (CE) loss and the logistic loss?

   A. The multi-category CE loss and the logistic loss are strongly connected.

   B. The logistic loss is not a convex loss, which is derived based on the maximum likelihood estimation.

   C. For classification problems, the CE loss is preferable to the mean-squared error (MSE) loss.

   D. Since the CE loss is a smooth function, we can use back-propagation to train neural networks with the CE loss.

4. (10 points) Which one is NOT an advantage of mini-batch stochastic gradient descent (SGD)?

   A. Mini-batch SGD is a deterministic algorithm.

   B. Mini-batch SGD exploits GPU hardware well.

   C. Mini-batch SGD is computationally efficient since it makes frequent updates to parameters.

   D. Mini-batch SGD can often escape local minima due to noisy gradients.

   ____

   Solution:

5. (10 points) Which one is NOT a common regularization technique in deep learning?

   A. Dropout.

   B. Weight decay.

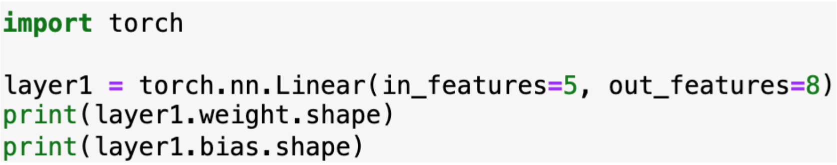
   C. Data splitting.

   D. Early stopping.

   ____

   Solution:

6. (10 points) [Dimension calculation] (i) Suppose that we create a linear layer in PyTorch as follows. What is the expected output?



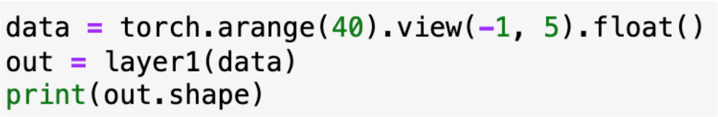
```
import torch

layer1 = torch.nn.Linear(in_features=5, out_features=8)
print(layer1.weight.shape)
print(layer1.bias.shape)
```

____

Solution:

(ii) Next, we create an input tensor and pass it through the linear layer. What is the expected output?

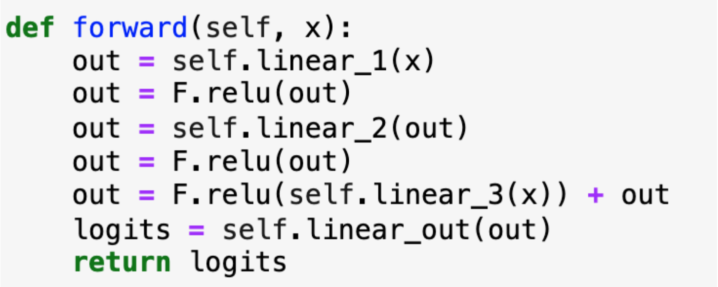


```
data = torch.arange(40).view(-1, 5).float()
out = layer1(data)
print(out.shape)
```

Solution:

7. (10 points) [*Translating code to math*] Suppose that in PyTorch we define a neural network class with three linear layers `linear_1`, `linear_2`, `linear_3`, and an output linear layer `linear_out`. All linear layers are created with NO bias terms. We use the symbols $W_1, W_2, W_3, W_{\text{out}}$ to represent the weight matrices of these linear layers respectively. We also use the symbol σ to represent the ReLU activation function.

   The following code gives the forward pass in the neural network class definition. Given an input x, can you write down the **mathematical expression** to represent the forward pass using the above symbols?



```
def forward(self, x):
    out = self.linear_1(x)
    out = F.relu(out)
    out = self.linear_2(out)
    out = F.relu(out)
    out = F.relu(self.linear_3(x)) + out
    logits = self.linear_out(out)
    return logits
```

Solution:

8. (10 points) Suppose that we have a multilayer perceptron (MLP). In one hidden layer, a 200-dimensional feature vector is mapped to a 32-dimensional feature vector. We initialize weights independent from a normal distribution $N(0, \sigma^2)$. According to He initialization, what is the value of σ? *Hint: the formula is* $\sigma = \sqrt{2/\text{width}}$.

   Solution:

9. (20 points) Pleaes explain in words what the following PyTorch code means.
   (i) `torch.manual_seed(42)`
   (ii) `features = features.view(-1, 28*28).to(‘cuda:0’)`
   (iii) `optimizer.zero_grad()`
   (iv) `train_loader = torch.utils.data.DataLoader(dataset=train_dataset,`
   `batch_size=64, shuffle=True)`
   (v) `net = torch.nn.Sequential(`
   `torch.nn.Linear(20, 30),`
   `torch.nn.ReLU(),`
   `torch.nn.Dropout(0.5),`
   `torch.nn.Linear(30, 10)`
   `)`

   Solution: