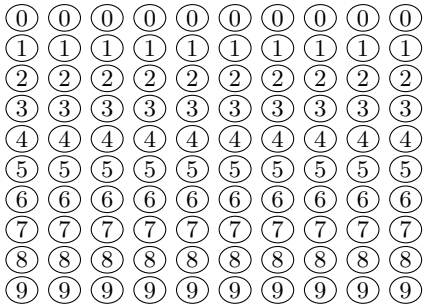
Last Name: _____

First Name: _____

Student Number:



APS360 Artificial Intelligence Fundamentals

Date : June 20, 2019
Duration: **110 minutes**
Aids Allowed: **None**
Instructor: Lisa Zhang

---

*Do **not** turn this page until you have received the signal to start.*
In the meantime, please read the instructions below *carefully*.

This test consists of 9 questions on 16 pages (including this one), printed on both sides of the paper. When you receive the signal to start, please make sure that your copy of the test is complete, fill in the identification section above. Bubble in your student number in either pen or pencil.

Answer each question directly on the test paper, in the space provided, using either a blue or black pen or a pencil. If you need more space for one of your solutions, use the extra pages at the end of the test paper and *indicate clearly the part of your work that should be marked*.

Write up your solutions carefully! Marks cannot be awarded for solutions that are not understandable by the grader, and may be deducted if you make false assertions. If you are giving only one part of an answer, indicate clearly what you are doing. Part marks might be given for incomplete solutions.

Marking Guide

1: _____/10
2: _____/ 4
3: _____/ 8
4: _____/12
5: _____/ 9
6: _____/11
7: _____/12
8: _____/ 7
9: _____/ 7

TOTAL: _____/80

Good Luck!

over...

Question 1. [10 MARKS]

Circle the best answer for each of the questions below. Do not circle more than one answer per question.

Part (a) [1 MARK]

For which of the following problems would you choose machine learning over a different technique?

(A) Determining whether a piece of Python code prints out the value "Hello, world".

(B) Determining whether a photograph is in black and white or in colour.

(C) Determining whether a photograph is of a young person or an old person.

(D) All of the above.

(E) Only (b) and (c).

Part (b) [1 MARK]

What is the approximate value of `softmax([1, 1, 1, 2])`?

(A) `[0.175, 0.175, 0.175, 0.475]`

(B) `[0.5, 0.5, 0.5, 1]`

(C) `[0.25, 0.25, 0.25, 0.50]`

(D) `[1, 1, 1, 2]`

(E) `[0.25, 0.25, 0.25, 0.25]`

Part (c) [1 MARK]

Which of the following is not an example of a distributed encoding (distributed representation)?

(A) The autoencoder embedding of an image.

(B) The AlexNet features of an image.

(C) The one-hot encoding of a word.

(D) The GloVe embedding of a word.

(E) All of the above are distributed encodings.

Part (d) [1 MARK]

Which of the following will most likely produce a more noisy training curve?

(A) Decreasing the batch size.

(B) Decreasing the learning rate.

(C) Decreasing the size of the training set.

(D) Increasing the size of the training set.

(E) Increasing the number of parameters of the neural network.

Part (e) [1 MARK]

Assuming a learning rate that is not too large, which of the following statements about training curves is true?

(A) If the batch size is 1, then the training loss will always decrease in each iteration.

(B) A larger batch size is typically required for larger inputs (e.g. larger images).

(C) If the batch size is the same as the size of the training set, then the training loss will always decrease in each iteration.

(D) Training an autoencoder requires a larger batch size than training a supervised learning classifier with a similar number of parameters.

(E) If the training accuracy is not decreasing, then neither is the training loss.

CONT'D...

Part (f) [1 MARK]

Which of the following helps prevent overfitting?

(A) Increasing the number of layers in a neural network.
(B) Training for more epochs.
(C) Using a larger batch size.
(D) Using a larger training set.
(E) Both (c) and (d).

Part (g) [1 MARK]

Which convolutional neural network architecture first introduced the idea of repeated sub-modules?

(A) LeNet
(B) AlexNet
(C) GoogLeNet
(D) ResNet
(E) None of the above

Part (h) [1 MARK]

Which of the following statements about regularization is true?

(A) It doesn't make sense to apply dropout to a neural network's output layer.
(B) It doesn't make sense to apply dropout to a neural network's input layer.
(C) Data augmentation can only be used for image inputs.
(D) Weight decay is usually more effective than transfer learning at preventing overfitting.
(E) Both (a) and (d) are true.

Part (i) [1 MARK]

Which of the following statements about autoencoders is true?

(A) The encoder must have the same number of parameters as the decoder for the autoencoder to learn reasonable representations.
(B) The decoder must have the more parameters than the encoder for the autoencoder to learn reasonable representations.
(C) If the encoder is fully-convolutional, then the decoder must also be fully-convolutional for the autoencoder to learn reasonable representations.
(D) A denoising autoencoder can have the exact same architecture as a normal autoencoder.
(E) Both (a) and (c) are true.

Part (j) [1 MARK]

Which of the following about word2vec and GloVe is true?

(A) Words with similar GloVe embeddings must have similar meanings.
(B) Training a word2vec model removes human biases in text.
(C) The word2vec architecture takes a sequence of variable length as input.
(D) All of the above are true.
(E) None of the above are true.

Question 2. [4 MARKS]

Part (a) [2 MARKS]

Sketch the ReLU activation function, clearly showing the approximate values of ReLU(-10), ReLU(0) and ReLU(10).

Part (b) [2 MARKS]

Sketch the tanh activation function, clearly showing the approximate values of tanh(-10), tanh(0) and tanh(10).

Question 3. [8 MARKS]

Part (a) [3 MARKS]

What is the purpose of each of the following data sets in the context of neural network training?

1. Training Set:

2. Validation Set:

3. Test Set:

Part (b) [2 MARKS]

Why do popular data sets like MNIST and CIFAR have a standard **test set**?

Part (c) [3 MARKS]

We discussed data augmentation techniques to help combat overfitting. Does it make sense to use data augmentation techniques to increase the size of the **test set**? Why or why not?

Question 4. [12 MARKS]

What is the expected output of the commands below? You may assume that all required packages like `torch` and `torch.nn` are imported. The commands are run in the same Python shell (or Colab notebook) from top to bottom. If you expect that an error would occur, write "ERROR" without further explanation.

```
>>> x1 = torch.randn(10, 16) # create a random tensor of shape [10, 16]
>>> x1.shape
torch.Size([10, 16])

>>> x2 = torch.randn(20, 3, 16, 16) # NCHW
>>> x2.shape
torch.Size([20, 3, 16, 16])

>>> layer = nn.Linear(in_features=16, out_features=32)
>>> layer(x1).shape
```

```
>>> conv = nn.Conv2d(in_channels=3, out_channels=7, kernel_size=5, padding=0)
>>> conv(x2).shape
```

```
>>> pool = nn.MaxPool2d(kernel_size=2, stride=2)
>>> pool(x2).shape
```

```
>>> conv3 = nn.Conv2d(in_channels=3, out_channels=7, kernel_size=5, padding=2,
                      stride=2)
>>> conv3(x2).shape
```

```
>>> convt = nn.ConvTranspose2d(in_channels=3, out_channels=1, kernel_size=5,
                               stride=2, padding=2, output_padding=0)
>>> convt(x2).shape
```

```
>>> convt2 = nn.ConvTranspose2d(in_channels=3, out_channels=1, kernel_size=5,
                                stride=2, padding=0, output_padding=1)
>>> convt2(x2).shape
```

Question 5. [9 MARKS]

Consider the task of classifying whether a 32×32 pixel greyscale image of an animal is of a cat, dog, horse, or bird.

Part (a) [6 MARKS]

Here is an incorrect implementation of a neural network model for performing the classification task. There are three issues with the code. Identify all issues, and propose a correction for each.

```
class AnimalClassifier(nn.Module):
    def __init__(self):
        super(AnimalClassifier, self).__init__()
        self.layer1 = nn.Linear(in_features=32, out_features=20)
        self.dropout = nn.Dropout(p=0.2)
        self.layer2 = nn.Linear(in_features=16, out_features=4)
    def forward(self, img):
        flattened = img.view(-1, 32 * 32)
        activation = self.layer1(flattened)
        activation = self.dropout(activation)
        return self.layer2(activation)
```

1.

2.

3.

Part (b) [3 MARKS]

What does `img.view(-1, 32 * 32)` in the `forward` method do, and why is it required? The PyTorch API states that the method `view` "returns a new tensor with the same data as the self tensor but of a different shape".

Question 6. [11 MARKS]

Part (a) [6 MARKS]

Complete the `train` function, which is a shell of the function we will use to train our `AnimalClassifier` from the previous question. Do so by filling the boxes with the letter that corresponds to the line of code given below. Fill each box with exactly one letter.

```
def train(model, data, batch_size, num_epoch):
    train_loader = torch.utils.data.DataLoader(data, batch_size=batch_size)

    optimizer = [ ]
    criterion = [ ]

    for epoch in range(num_epochs):
        for images, labels in iter(train_loader):
            optimizer.zero_grad()

            output = [ ]
            loss = [ ]
            [ ]
            [ ]
```

A. `criterion(images, labels)`
B. `criterion(outputs, labels)`
C. `criterion(torch.sigmoid(outputs), labels)`
D. `criterion(torch.softmax(outputs, dim=1), labels)`
E. `loss.backward()`
F. `model(images)`
G. `model(images, hidden)`
H. `model(images, target)`
I. `model(images, target, hidden)`
J. `model(target, hidden)`
K. `nn.BCEWithLogitsLoss()`
L. `nn.CrossEntropyLoss()`
M. `nn.MSELoss()`
N. `optim.SGD(model.parameters(), lr=0.01, momentum=0.9)`
O. `optim.SGD(model.parameters(), lr=100, momentum=0.9)`
P. `optimizer(outputs, labels)`
Q. `optimizer(torch.sigmoid(outputs), labels)`
R. `optimizer(torch.softmax(outputs, dim=1), labels)`
S. `optimizer.step()`
T. `optimizer.zero_grad()`

Part (b) [1 MARK]

You have trained the `AnimalClassifier` model, and are now using it to make predictions about a test image called `image`. Which of the following pieces of code correctly computes the prediction probabilities?

(a) `model(image)`

(b) `torch.sigmoid(model(image, ground_truth_label))`

(c) `torch.softmax(model(image, ground_truth_label), dim=1)`

(d) `torch.tanh(model(image))`

(e) `torch.sigmoid(model(image))`

(f) `torch.sigmoid(torch.softmax(model(image)))`

(g) `torch.softmax(model(image), dim=1)`

Part (c) [2 MARKS]

You run the code from part (b) two separate times on the same image, and received two different prediction probability distributions. Why might this be?

Part (d) [2 MARKS]

Write down the piece of necessary code that was likely omitted. (If you don't remember the code, that's okay. Write down what the code is intended to accomplish to receive full marks.)

Question 7. [12 MARKS]

Consider the following implementation of an autoencoder, to be trained to generate MNIST images. Recall that the MNIST dataset contains 28x28 pixel greyscale images of hand written digits.

```
class Autoencoder(nn.Module):
    def __init__(self):
        super(Autoencoder, self).__init__()
        self.encoder = nn.Sequential(
            nn.Conv2d(in_channels=1, out_channels=16, kernel_size=3, stride=2, padding=1),
            nn.ReLU(),
            nn.Conv2d(in_channels=16, out_channels=32, kernel_size=3, stride=2, padding=1),
            nn.ReLU(),
            nn.Conv2d(in_channels=32, out_channels=64, kernel_size=7, padding=0)
        )
        self.decoder = nn.Sequential(
            nn.ConvTranspose2d(in_channels=64, out_channels=32, kernel_size=7, padding=0),
            nn.ReLU(),
            nn.ConvTranspose2d(in_channels=32, out_channels=16, kernel_size=3,
                               stride=2, padding=1, output_padding=1),
            nn.ReLU(),
            nn.ConvTranspose2d(in_channels=16, out_channels=1, kernel_size=3,
                               stride=2, padding=1, output_padding=1),
            nn.Sigmoid()
        )
    def forward(self, x):
        return self.decoder(self.encoder(x))
```

Part (a) [6 MARKS]

Compute the total number of parameters in this model's **encoder only**. Show all your work, but don't perform the multiplications.

Part (b) [3 MARKS]

Normally, we apply an activation function after a fully-connected or a convolutional layer. Why do we *not* apply an activation function to the output of the encoder before passing it to the decoder?

Part (c) [3 MARKS]

Suppose that we run the following code to generate a new MNIST digit from a random point in the embedding space.

```
# model = a trained Autoencoder model
emb = torch.randn(1, 64, 1, 1) # Returns a tensor filled with random numbers from
                               # a normal distribution with mean 0 and variance 1
new_img = model.decoder(emb)
```

Would you expect the generated image to look like a MNIST digit? Why or why not?

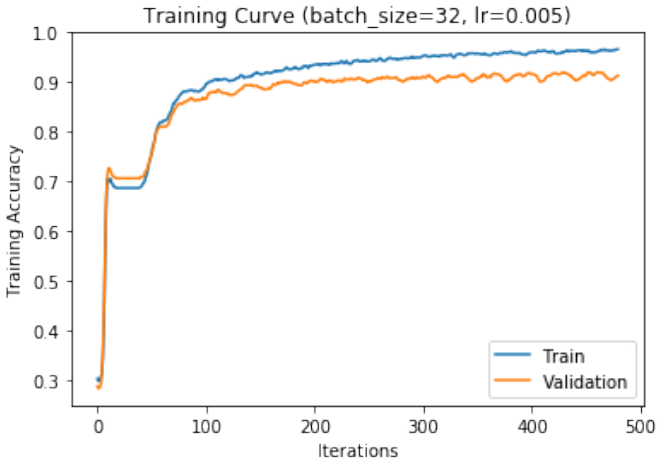
Question 8. [7 MARKS]

Part (a) [2 MARKS]

How many **backward passes** is required to train a neural network on a training set of 10,000 images for 10 epochs, using a batch size of 100?

Part (b) [3 MARKS]

Consider this training curve from the first few weeks of lecture, where we trained a neural network to determine whether an MNIST image contained either of the digits 0, 1 or 2. Explain the strange shape of the training curve in the first 100 iterations of training.



Part (c) [2 MARKS]

Is there evidence of overfitting in this training curve below? If so, how severe? Explain.

CONT'D...

Question 9. [7 MARKS]

Part (a) [3 MARKS]

In the transfer learning task in lab 3, we used the activations of the last AlexNet convolutional layer as input to our classifier. Why did we use the activation of the **last** convolutional layer, as opposed to the activations of an earlier convolutional layer?

Part (b) [2 MARKS]

During class, we saw that words with the most similar GloVe embedding to names (e.g. "Michael") are other names (e.g. "Chris"). Explain why this is the case.

Part (c) [2 MARKS]

Suppose that a neural network is used to decide whether a loan application is approved or rejected. Will decisions made by the neural network be less biased than those made by a human? Why or why not?

Additional page for answers

Additional page for answers

Additional page for answers