

APS360 Artificial Intelligence Fundamentals

Instructor: Lisa Zhang
Date : February 28, 2019
Duration: **110 minutes**
Aids Allowed: **None**

Student Number: _____

Family Name(s): _____

Given Name(s): _____

*Do **not** turn this page until you have received the signal to start.*

In the meantime, please read the instructions below carefully.

MARKING GUIDE

This test consists of 8 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, and write your name on the back of the last page.

Answer each question directly on the test paper, in the space provided. 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! If you are giving only one part of an answer, indicate clearly what you are doing. Part marks might be given for incomplete solutions where it is clearly indicated what parts are missing.

You **must** write the test in pen if you would like to potentially request for the test to be regraded.

1: _____/10

2: _____/10

3: _____/ 6

4: _____/10

5: _____/12

6: _____/15

7: _____/ 7

8: _____/10

TOTAL: _____/80

Midterm Test

Question 1. [10 MARKS]

Circle the best answer for each of the questions below.

Part (a) [1 MARK]

Which of the following is the **least** plausible trend in a training curve? (You may assume that a neural network is being trained to solve a classification problem).

- (A) The training loss increases with additional training.
- (B) The training accuracy decreases with additional training.
- (C) The training loss decreases with additional training.
- (D) The validation loss increases with additional training.
- (E) The validation loss decreases with additional training.

Part (b) [1 MARK]

Which of the following does **not** evaluate to 0?

- (A) `sigmoid(0)`
- (B) `tanh(0)`
- (C) `relu(-5)`
- (D) `relu(0)`

Part (c) [1 MARK]

Dropout is an example of ...

- (A) A way to prevent overfitting.
- (B) An optimizer.
- (C) A loss function.
- (D) A data augmentation technique.
- (E) A neural network architecture.

Part (d) [1 MARK]

Which of the following is **not** an example of unsupervised learning?

- (A) Identifying names and places in previously-published news articles.
- (B) Estimating housing prices using historical data.
- (C) Using historical traffic data to cluster drivers based on their behaviour.
- (D) Learning a set of word embeddings by predicting the context in which a word appears.
- (E) Both (b) and (c).

Part (e) [1 MARK]

The number of parameters of a neural network increase if we ...

- (A) Increase the embedding size of an autoencoder with only fully-connected layers.
- (B) Increase the size of our input image in a fully-convolutional network.
- (C) Decrease the length of our input sequence in a recurrent neural network.
- (D) Decrease the size of our input image in a fully-convolutional network.
- (E) Increase the length of our input sequence in a recurrent neural network.

Part (f) [1 MARK]

You are training a model to determine whether a tweet is in English, French, or German. Which loss function would you use?

- (A) `optim.SGD`
- (B) `nn.BCEWithLogitsLoss`
- (C) `nn.MSELoss`
- (D) `nn.CrossEntropyLoss`
- (E) Both (a) and (d)

Part (g) [1 MARK]

Which of the following PyTorch layers have an output tensor with **smaller** width and height dimensions than the input tensor?

- (A) `nn.MaxPool2d` with stride set to 1.
- (B) `nn.Conv2dTranspose` with stride set to 2.
- (C) `nn.Conv2d` with stride set to 2.
- (D) None of the above.
- (E) Two or more of the above.

Part (h) [1 MARK]

Which of the following convolutional neural network architectures has the **most** number of layers?

- (A) AlexNet
- (B) VGG-19
- (C) GoogleLeNet
- (D) LeNet
- (E) ResNet

Part (i) [1 MARK]

Why do we need to pad sequences in an RNN?

- (A) We don't. Padding is for fully-convolutional networks.
- (B) We don't. Padding is for convolutional networks with some fully-connected layers.
- (C) To ensure all input sequences in a mini-batch have the same length.
- (D) To ensure all input sequences in the training set have the same length.
- (E) To ensure all input sequences in the validation set have the same length.

Part (j) [1 MARK]

Consider embedding a tweet by **summing** the GloVe embeddings of each word in a tweet. Which of the following is **true** about the embeddings of the tweets “i am happy” and “i am happy i am happy i am happy”?

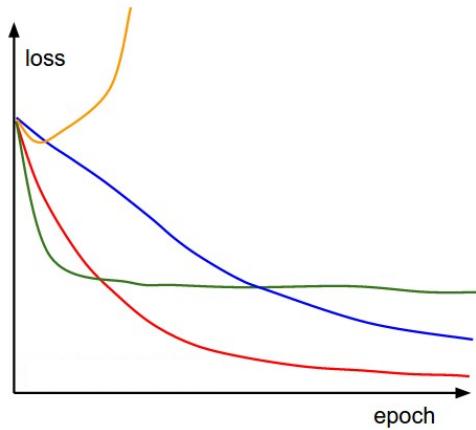
- (A) The embeddings are equal.
- (B) The embeddings are close when measured using the Euclidean distance, but not the cosine similarity.
- (C) The embeddings are close when measured using the cosine similarity, but not the Euclidean distance.
- (D) The embeddings are close when measured using either the Euclidean distance or the cosine similarity.
- (E) None of the above.

Midterm Test

Question 2. [10 MARKS]

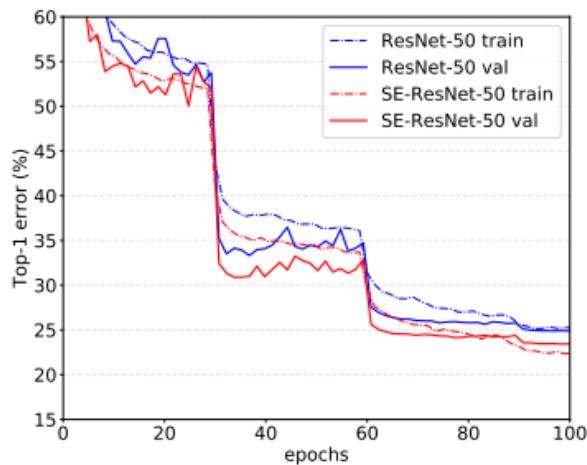
Part (a) [4 MARKS]

Which of the curves correspond to a learning rate that is (a) very high, (b) high, (c) good, and (d) low?
Label the four curves.



Part (b) [2 MARKS]

Describe what setting change in epochs 25 and 60 could have produced this training curve. Be brief.



Part (c) [4 MARKS]

Beginning week 4, we constructed data loaders like this:

```
torch.utils.data.DataLoader(train, batch_size=32, shuffle=True)
```

The PyTorch documentation for the parameter `shuffle` says that setting `shuffle=True` will “have the data reshuffled at every epoch”. What does this mean? What happens if we do not set `shuffle=True`, like we did in the first 3 weeks of the class? Sketch a training curve to describe the effect of this setting.

Midterm Test

Question 3. [6 MARKS]

Consider the following incorrect training code, where `model` is a neural network we wish to train, `optimizer` is an optimizer, `criterion` is a loss function, and `train_loader` is a `DataLoader` containing the training data.

```
for xs, ys in train_loader:  
    pred = model(xs)  
    loss = criterion(pred, ys)  
    optimizer.step()  
    loss.backward()
```

Part (a) [1 MARK]

Circle and label the line(s) of code that perform(s) the **forward pass**.

Part (b) [1 MARK]

Circle and label the line(s) of code that perform(s) the **backward pass**.

Part (c) [4 MARKS]

There are two issues with the above code. What are they?

1.

2.

Question 4. [10 MARKS]**Part (a)**

How many parameters are in the PyTorch layer:

```
nn.Conv2d(in_channels=5, out_channels=10, kernel_size=3)?
```

Part (b)

How many parameters are in the PyTorch layer:

```
nn.Conv2d(in_channels=5, out_channels=10, kernel_size=1, stride=2)?
```

Part (c)

How many parameters are in the PyTorch layer `nn.MaxPool2d(kernel_size=2, stride=1)`?

Part (d)

How many parameters are in the PyTorch layer `nn.Linear(in_features=30, out_features=10)`?

Part (e)

How many parameters are in the PyTorch layer `nn.Dropout(p=0.2)`?

Midterm Test

Question 5. [12 MARKS]

The following (incorrectly implemented) model predicts whether an email message is spam. An email is represented as a sequence of words. Each word is represented using a unique integer identifier, used to look up its GloVe embedding.

```
class SpamClassifier(nn.Module):
    def __init__(self, input_size, hidden_size, glove, num_classes=2):
        super(SpamClassifier, self).__init__()
        self.input_size = input_size
        self.hidden_size = hidden_size
        self.emb = nn.Embedding.from_pretrained(glove.vectors)
        self.rnn = nn.LSTM(input_size, hidden_size, batch_first=True)
        self.fc = nn.Linear(hidden_size, num_classes)

    def forward(self, seq):
        embeddings = self.emb(seq)
        h0 = torch.zeros(1, embeddings.size(0), self.input_size)
        out, _ = self.rnn(embeddings, h0)
        out = self.fc(out)
        return out
```

Part (a) [6 MARKS]

Identify three issues with the code above.

1.

2.

3.

Part (b) [2 MARKS]

Suppose we decide to use a 50-dimensional GloVe vector for our `SpamClassifier`. What restrictions does that place on the values of `input_size` and `hidden_size`?

Part (c) [2 MARKS]

Why is it advantageous to use GloVe embeddings?

Part (d) [2 MARKS]

List one alternative to using GloVe embeddings in the recurrent neural network, and one of its drawbacks.
(If you provide multiple answers, only the first will be marked.)

Midterm Test

Question 6. [15 MARKS]

You would like to train a neural network classifier to detect pictures of your face. You want to use this model to unlock your phone by taking a picture to verify your identity (or a few pictures at slightly different angles). The problem is you have very few pictures of yourself that you can use for training.

Part (a) [3 MARKS]

List 3 techniques that you can use to augment your training data. (Do not list more than three.)

1.

2.

3.

Part (b) [3 MARKS]

What other technique can you use to drastically improve your neural network performance?

Part (c) [3 MARKS]

When developing neural network code, why is it a good idea to first see if your model can overfit a small set of training examples?

Part (d) [2 MARKS]

Would this overfit model have high **validation** accuracy? Explain.

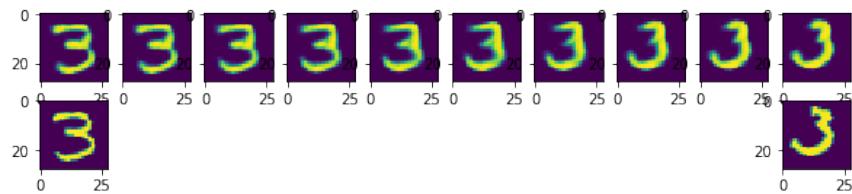
Part (e) [4 MARKS]

What is **checkpointing** in the context of neural network training, and why should you checkpoint?

Midterm Test

Question 7. [7 MARKS]

Explain how we obtained these images in class.



Question 8. [10 MARKS]**Part (a)** [5 MARKS]

PyTorch implementation of weight decay penalizes large parameters of all types. Why might we want to allow large **biases**, and only decay neural network **weights**?

Part (b) [5 MARKS]

How it is possible for a neural network to overfit to the validation set.

Midterm Test

Additional page for answers

Additional page for answers

On this page, please write nothing except your name.

Family Name(s): _____

Given Name(s): _____