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.

This test consists of 5 questions on 8 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.

Marking Guide



Midterm Practice Questions

Question 1. [5 MARKS]

Circle the best answer for each of the questions below.

Part (a) [1 MARK]

Batching is straightforward in all but the following type of network:

- (A) Word2Vec
- (B) Fully Convolutional neural networks
- (C) Feed-forward neural networks
- (D) Convolutional neural networks
- (E) Recurrent neural networks

Part (b) [1 MARK]

Which of the following does **not** help reduce overfitting?

Part (c) [1 MARK]

Consider the following PyTorch code: x = img.view(-1, 28 * 28). What shape does the tensor img have to be for the code to throw an error?

Part (d) [1 MARK]

A transpose convolution is used in the following context:

Part (e) [1 MARK]

Which of the following is **true** about recurrent neural networks?

- (A) Using dropout layers.
- (B) Using weight decay to penalize large weights.
- (C) Augmenting the test data by adding small, random perturbations.
- (D) Stopping training when the validation accuracy stops improving.
- (E) Augmenting the training data by adding small, random perturbations.
- (A) [1, 28, 28]
- (B) [10, 14, 56]
 - (C) [4, 784]
- (D) [28, 28, 28]
- (E) None of the above
- (A) A pixel-wise prediction problem.
- (B) A recurrent neural network that takes images as input.
- (C) An image autoencoder.
- (D) All of the above.
- (E) Only (a) and (c).
- (A) The size of the RNN output vector is the same as the size of the hidden state.
- (B) The size of the RNN output vector is the same as the size of the input embedding.
- (C) The PyTorch module nn.RNN is more often used than nn.LSTM.
- (D) The PyTorch module $\tt nn.LSTM$ uses weight sharing, and thus has fewer parameters than $\tt nn.RNN.$
- (E) The size of the hidden state is the same as the size of the input embedding.

Question 2. [5 MARKS]

Explain what could cause the training curve to have the below shape.



Question 3. [12 MARKS]

Consider the following two networks NetworkA and NetworkB:

```
class NetworkA(nn.Module):
   def __init__(self):
        super(NetworkA, self).__init__()
        self.conv1 = nn.Conv2d(in_channels=3, out_channels=5, kernel_size=3, padding=1)
        self.pool = nn.MaxPool2d(2, 2)
        self.conv2 = nn.Conv2d(in_channels=5, out_channels=10, kernel_size=3, padding=1)
       self.fc = nn.Linear(10 * 5 * 5, 10)
    def forward(self, x):
        x = self.pool(F.relu(self.conv1(x)))
        x = self.pool(F.relu(self.conv2(x)))
        x = x.view(-1, 10 * 5 * 5)
        x = self.fc(x)
        x = x.squeeze(1) # Flatten to [batch_size]
        return x
class NetworkB(nn.Module):
    def __init__(self):
       super(NetworkA, self).__init__()
        self.fc1 = nn.Linear(20 * 20, 300)
        self.fc2 = nn.Linear(300, 100)
        self.fc3 = nn.Linear(100, 10)
    def forward(self, x):
       x = x.view(-1, 20 * 20)
        x = F.relu(self.fc1(x))
       x = F.relu(self.fc2(x))
       x = self.fc3(x)
        return x
```

Part (a) [5 MARKS]

How many parameters are in NetworkA?

Part (b) [5 MARKS] How many parameters are in NetworkB?

Part (c) [2 MARKS]

Which of the two networks is more likely to overfit?

Question 4. [6 MARKS]

Consider the following neural network model written in PyTorch. Identify three issues with the code.

```
class Classifier(nn.Module):
    def __init__(self):
        super(Classifier, self).__init__()
        self.layer1 = nn.Linear(28 * 28, 40)
        self.layer2 = nn.Linear(30, 5)
    def forward(self, img):
        flattened = img.view(-1, 28 * 28)
        activation1 = self.layer1(flattened)
        activation2 = self.layer2(activation1)
        return torch.softmax(activation2)
```

1.

2.

3.

Midterm Practice Questions

Question 5. [12 MARKS] Part (a) [4 MARKS] Describe the architecture of a word2vec model.

Part (b) [4 MARKS]

Describe two possible subsampling methods used in convolutional neural networks.

Part (c) [4 MARKS]

Suppose you are training an RNN to determine the gender of the author of a tweet, but you have a small training set. What techniques can you use?

Additional page for answers

On this page, please write nothing except your name.

Family Name(s):

Given Name(s):