Sample Test | Deep Learning, Spring 2019 | February 21, 2019 | Total Points = 60

All questions have equal points. This is a closed book exam. No electronics or cheat sheets are allowed.

Name:

---

1. Assume that the variable "YVALIDATION" has the true labels (1s and 0s) for the Pima Indians Diabetes Dataset and "prediction" variable contains the predictions from a trained neural network model. Given the following code, calculate the precision, accuracy, and recall.

```
print(YVALIDATION[0:10])

prediction = model.predict(XVALIDATION)
print('')
print(prediction[0:10])
print ('')
print(prediction[0:10].round())

accuracy = accuracy_score(YVALIDATION[0:10], prediction[0:10].round())
precision = precision_score(YVALIDATION[0:10], prediction[0:10].round())
recall = recall_score(YVALIDATION[0:10], prediction[0:10].round())

print("Accuracy: %.2f%%" % (accuracy * 100.0))
print("Precision: %.2f%%" % (precision * 100.0))
print("Recall: %.2f%%" % (recall * 100.0))
```

```
[1.00 0.00 1.00 0.00 1.00 0.00 1.00 0.00 1.00 1.00]

[[0.70]
 [0.17]
 [0.70]
 [0.13]
 [0.62]
 [0.23]
 [0.18]
 [0.35]
 [0.62]
 [0.38]]

[[1.00]
 [0.00]
 [1.00]
 [0.00]
 [1.00]
 [0.00]
 [0.00]
 [0.00]
 [1.00]
 [0.00]]
```

2. Of the two models (model1 and model2) below, which is for regression and which is for binary classification? Explain.

```
model1 = Sequential()
model1.add(Dense(12, input_dim=8, activation='relu'))
model1.add(Dense(8, activation='sigmoid'))
model1.add(Dense(1, activation='linear'))

model2 = Sequential()
model2.add(Dense(12, input_dim=8, activation='sigmoid'))
model2.add(Dense(8, activation='relu'))
model2.add(Dense(1, activation='sigmoid'))

```

3. The code below uses a simple CNN architecture to train a model on the standard MNIST digits dataset (0 to 9 digits).

Complete the code below to have the following architecture:
(a) 16 convolutional filters in the first layer, each of size 3 x 3, with sigmoid activations
(b) 4 convolutional filters in the second layer, each of size 3x3, with sigmoid activations
(c) A dense layer as output layer

```
from keras.datasets import mnist
from keras.utils import to_categorical
from keras import layers, models

(train_images, train_labels), (validation_images, validation_labels) = mnist.load_

train_images = train_images.reshape( ( 60000, 28, 28, 1 ) )
train_images = train_images.astype( 'float32' ) / 255
validation_images = validation_images.reshape( ( 10000, 28, 28, 1 ) )
validation_images = validation_images.astype( 'float32' ) / 255

train_labels = to_categorical( train_labels )
validation_labels = to_categorical( validation_labels )

model = models.Sequential()
model.add(layers.Conv2D(                                                    ))
model.add(layers.Conv2D(                                                    ))
model.add(                                                                   )
model.add(layers.Dense(                                                     ))

model.compile( optimizer = 'rmsprop', loss = 'categorical_crossentropy', metrics =
model.fit( train_images, train_labels, epochs = 1, batch_size = 256 )
```

4. The output below shows the summary of a convolutional neural network constructed. Answer the following referring to the output:

(a) How many filters are there in the first convolutional layer?
(b) How is the total number of parameters in the first layer 160?
(c) How is the total number of parameters in the second convolutional layer 580?

```
from keras.datasets import mnist
from keras.utils import to_categorical
from keras import layers, models

```

```
model = models.Sequential()
...
model.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_3 (Conv2D)            (None, 26, 26, 16)        160
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 24, 24, 4)         580
_________________________________________________________________
flatten_2 (Flatten)          (None, 2304)              0
_________________________________________________________________
dense_59 (Dense)             (None, 10)                23050
=================================================================
Total params: 23,790
Trainable params: 23,790
Non-trainable params: 0
_________________________________________________________________
```

5. What will be the output of the following code?

```
import numpy as np
x = np.array([[1,2],[3,4]])
print np.sum(x)
print np.sum(x, axis=0)
print np.sum(x, axis=1)
```

6. What will be the output of the following code?

```
import numpy as np
x = np.array([[1,2,3], [4,5,6], [7,8,9], [10, 11, 12]])
v = np.array([1, 0, 1])
y = x + v
print (x)
print (v)
print (y)
```