**ECE 5470 Classification, Machine Learning, and Neural Network Review**

**Due December 1.**

Instructions:

These questions are to be answered on this document which should be submitted to blackboard in pdf format. This document is made available in both word (.docx) and pdf formats. The answers may be added to the document in many ways including handwritten and scanned. Converting the final answers to a pdf format for submission is the responsibility of the student.

For questions that ask for an explanation or definition it should be able provide a good response in one to three sentences in the space provided following the questions. For numerical questions shoe your working.

1. Suppose your input is 16x16 color (RGB) image, and you use a 3x3 convolutional layer with 30 neurons. What are the total number of weights in the hidden layer?
2. You have an input volume that is 63x63x16 and convolve it with 3 filters that are each 7x7, and stride of 1. You want the (“same convolution” setting in TensorFlow). How large is the padding?
3. Which of the following statement is true about k-NN algorithm?
4. k-NN performs much better if all of the data have the same scale
5. k-NN works well with a small number of input variables (p), but struggles when the number of inputs is very large
6. k-NN makes no assumptions about the functional form of the problem being solved

a) 1 and 2
b) 1 and 3
c) Only 1
d) All of the above
f) None of the abov

1. A logistic regression classifier, which identifies an abject as one of 4 classes (a-d) , provides individual probability estimates for an image of a=0.4, b=.0.2, c=0.9, and d=0.8
	* 1. Which class is the object identified as?
		2. what is the probability of correct classification?
2. What is data augmentation and how does it improve classifier performance.
3. Below is a CNN model specified in Keras

input\_shape = (1, 32, 128)

model = Sequential()

model.add(Conv2D(32, kernel\_size=(3, 3),

 activation='relu',

 input\_shape=input\_shape))

model.add(Conv2D(64, (3, 3), activation='relu'))

model.add(MaxPooling2D(pool\_size=(2, 2)))

model.add(Dropout(0.25))

model.add(Flatten())

model.add(Dense(128, activation='relu'))

model.add(Dropout(0.5))

model.add(Dense(num\_classes, activation='softmax'))

model.compile(loss=keras.losses.categorical\_crossentropy,

 optimizer=keras.optimizers.Adadelta(),

 metrics=['accuracy'])

Draw a block diagram of the model and clearly mark which block have trainable weights how they are organized; e.g., 12 x 16 x 3.

1. In a two-class task the results from logistic regression classifier are as follows:

(sorted in order of probability of correct class)

|  |  |  |
| --- | --- | --- |
| Index | Confidence positive | Correct class |
| 1 | .97 | + |
| 2 | .93 | + |
| 3 | .72 | - |
| 4 | .51 | - |
| 5 | .45 | + |
| 6 | .33 | - |
| 7 | .25 | - |
| 8 | .20 | - |

a. Carefully sketch or plot the ROC graph.

Indicate the value of each point on the graph.

Show working.

b. Draw and label the confusion matrix.

c. Extra credit: carefully draw the Precision Recall Curve (PRC)

1. For the images used in in the [Lab 7 tutorial](http://www.via.cornell.edu/courses/ece547/lab/lab7/lab7a.html), how are they preprocessed before being input to the classifier? List each preprocessing step and its purpose:
2. How is a ResNet convolutional layer different than a traditional CNN layer?
3. What is the advantage of a Reset layer compared to a traditional CNN layer?