

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

### Due December 1. **Solution set**

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 show 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?

Every neuron generates one pixel value in the output volume, it calculates the convolution value for one pixel point, so one neuron has  $3 \times 3 = 9$  weights + a bias. The total weights in the hidden layer is  $30 \times (9+1) = 300$ .

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?

Zero pad each input feature map (each depth of the input volume) by 3

((filter size - 1)/2) pixels over all four directions.

(Number of padding pixels:  $(63 \times 3 \times 4 + 4 \times 3 \times 3) \times 16 = 12672$ )

3. Which of the following statement is true about k-NN algorithm?
  1. k-NN performs much better if all of the data have the same scale
  2. k-NN works well with a small number of input variables (p), but struggles when the number of inputs is very large
  3. 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 above

4. A logistic regression classifier, which identifies an object 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
- a. Which class is the object identified as?

C

This was not the best formed question; better wording would have been: "If we use this classifier with a 1-out-of-n code output (i.e., as a multinomial logistic regression)"

- b. what is the probability of correct classification?

For the individual outputs we have  $p_{i,j} = \frac{1}{1 + e^{-f_{i,j}}}$

Thus  $e^{f_{i,j}} = \frac{p_{i,j}}{1-p_{i,j}}$

$e^{f_{i,a}} = 0.666$ ,  $e^{f_{i,b}} = 0.25$ ,  $e^{f_{i,c}} = 9$ ,  $e^{f_{i,d}} = 4$

then the class probability is given by the softmax function

Thus, the probability of correct classification is

$p(\text{correct}) = 9 / (0.666 + 0.25 + 9 + 4) = 0.647$

$$p_{i,j} = \frac{e^{f_{i,j}}}{\sum_k e^{f_{i,k}}}$$

5. What is data augmentation and how does it improve classifier performance.

The concept that the effective size of the training set may be increased by augmenting it with distorted or modified version of the original training images to improve classifier training (reduce overfitting)

Data augmentation may add value to base data by adding information derived from internal and external sources (for instance, we could enlarge the image dataset by rotating, translating, shrinking original training images). It may make the dataset more "noisy" with the goal of achieving a more robust model (prevent overfitting).

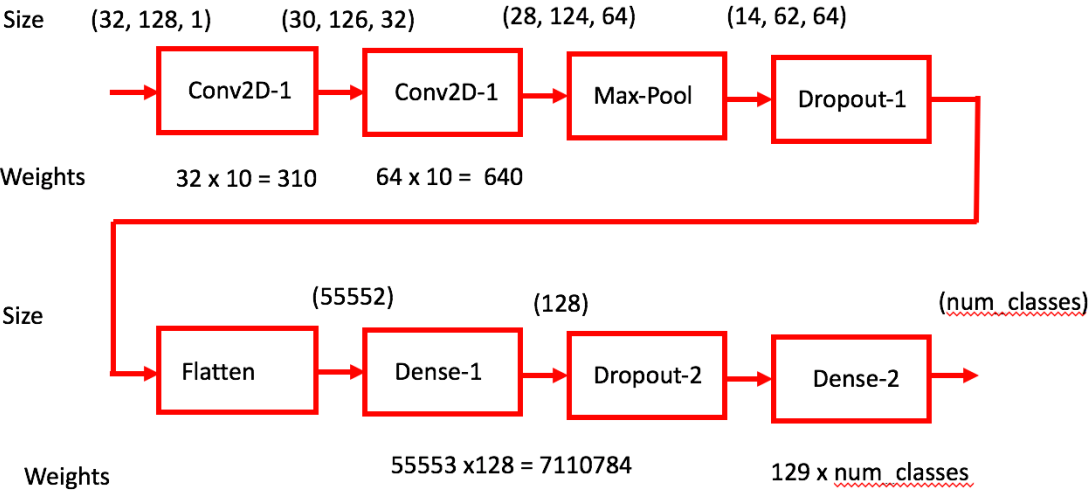
6. Below is a CNN model specified in Keras

```
input_shape = (32, 128, 1)
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.

See solution on next page

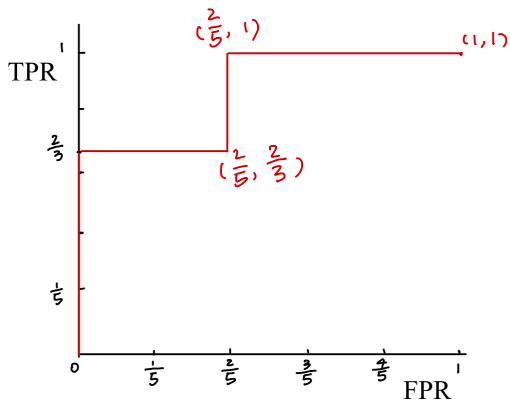


Keras defaults: use\_bias=True, padding="valid" (none)

7. 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.

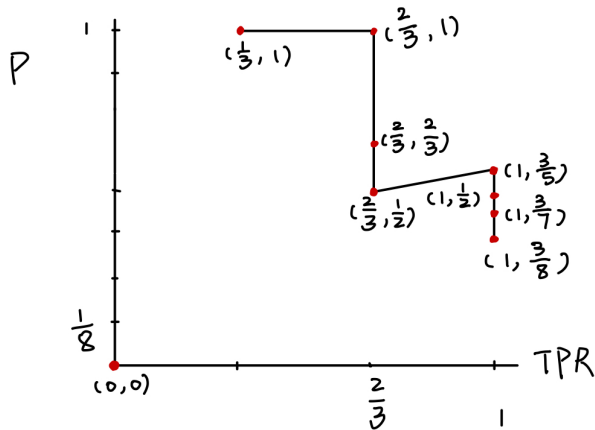


b. Draw and label the confusion matrix.

Supposed the threshold is 0.5:

		actual class	
		Positive	Negative
predicted class	Positive	(TP) 2	(FP) 2
	Negative	(FN) 1	(TN) 3

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



8. For the images used in in the [Lab 7 tutorial](#), how are they preprocessed before being input to the classifier? List each preprocessing step and its purpose:

(MNIST preprocessing)

1. Rescale binary images into a resampled 20 x 20 bounding box.      Size normalization
2. Move the COM to the center of a 28 x 28 image.      Location normalization
3. Rescale the data to have values between 0 and 1.      Pixel scale normalization
4. Standardize the training and testing data to have mean 0 and standard deviation 1.  
     --- Pixel scale normalization (makes step 3 unnecessary)

9. How is a ResNet convolutional layer different than a traditional CNN layer?

From the paper “Deep Residual Learning for Image Recognition”, ResNet implements shortcut connections to perform identity mapping, and the identity mapping outputs are added to the outputs of the stacked layers. A traditional CNN simply stacks layers. That is, after a nonlinear stage the inputs are added to outputs before an activation function.

10. What is the advantage of a ResNet layer compared to a traditional CNN layer?

As stated in the paper “Deep Residual Learning for Image Recognition”

1. ResNet is easier to optimize compared to traditional CNN layers.
2. ResNet gains accuracy from greatly increased depth, while traditional CNNs encounter degradation when networks’ depth increases.

One motivation for skipping over layers in CNNs is to avoid the problem of vanishing gradients by reusing activations from a previous layer until the layer next to the current one has learned its weights. This may facilitate the more rapid practical training of deeper networks.