Artificial Neural Networks and Deep Learning

Prof. Matteo Matteucci and Prof. Giacomo Boracchi
Date: 10/02/2020

Name:

Surname:

Student id (matricola):

Notes: Write ***clearly*** your answers in the provided sheets we need to read in order to grade! Write your name, surname and id, **on any** sheet of paper you turn in. You **can** use pencil, although we do not care if you just erase with a strikethrough. You **cannot** use books, notes, slides, cheatsheets, and electronic devices of any sort. For each answer you provide, please state clearly the number of the exercise and the number of the question, e.g., *Ex.1.2*, which you are currently answering.

Exercise 1 (1 + 2 + 2 = 5 points):

With reference to Feed Forward Neural Networks answer the following questions:

1. What is the difference between supervised and unsupervised learning?
2. Make an example of neural network used for supervised learning, i.e., describe the problem faced, the model, and the loss function used
3. Make an example of neural network used for unsupervised learning, i.e., describe the problem faced, the model and the loss function used

Exercise 2 (1 + 2 + 2 + 2 = 7 points):

Many architectural details are involved in obtaining a successful neural network model. For each of the following, list and discuss *briefly* the available options:

1. Output activation function
2. Loss function
3. Weight initialization
4. Regularization

Exercise 3 (1 + 3 + 2 + 1 + 2 = 9 points):

Consider the following Python snippet

```
import keras
from keras.utils  import np_utils
from keras.models import Sequential, Model
from keras.layers import Dense, Activation, Flatten, Conv2D,
from keras.layers import MaxPooling2D, BatchNormalization

pool_size          = (2,2) # size of pooling area for max pooling
nb_filters         = 64

model = Sequential()

model.add(Conv2D(nb_filters, (11,11), input_shape=(64, 64, 3), padding = "same"))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size = pool_size))
model.add(Conv2D(nb_filters*2, (5,5), padding = "same")) #2nd conv Layer starts
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size = pool_size))
model.add(Conv2D(nb_filters*4, (3,3), padding = "same")) #3rd conv Layer starts
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size = pool_size))
model.add(Conv2D(nb_filters, (1,1), padding = "same"))   #4th conv Layer starts
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size = pool_size))
model.add(Flatten())
model.add(Dense(64))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(Dense(16))
model.add(Activation('relu'))
model.add(Dense(21))
model.add(Activation('softmax'))

model.summary()
```

Answer the following questions providing a short explanation for each

1. Is this a classification or a regression network? How many output?
2. The following is the output of the model.summary() command. A few numbers have been replaced by dots. Please, fill all the numbers in and in particular indicate which computation you are performing to get to each number. Note: the first dimension is the batch size, and as such this is set to None.
3. How big is the receptive field of the pixel at the center of the feature map after the first convolutional layer? How about the pixel at the center of the second convolutional layer?
4. This network has relatively large filters in the first layers. Can we replace that layer with others to reduce the number of parameters and at the same time preserve (or increase) the receptive field?

5. What happens if we remove the forth convolutional layer: does the number of parameters increase or decrease? Any comments about that layer?

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None,...,...,...)        .......
-----------------------------------------------------------------
activation_1 (Activation)    (None,...,...,...)        .......
-----------------------------------------------------------------
max_pooling2d_1 (MaxPooling2 (None,...,...,...)        .......
-----------------------------------------------------------------
conv2d_2 (Conv2D)            (None,...,...,...)        .......
-----------------------------------------------------------------
activation_2 (Activation)    (None,...,...,...)        .......
-----------------------------------------------------------------
max_pooling2d_2 (MaxPooling2 (None,...,...,...)        .......
-----------------------------------------------------------------
conv2d_3 (Conv2D)            (None,...,...,...)        .......
-----------------------------------------------------------------
activation_3 (Activation)    (None,...,...,...)        .......
-----------------------------------------------------------------
max_pooling2d_3 (MaxPooling2 (None,...,...,...)        .......
-----------------------------------------------------------------
conv2d_4 (Conv2D)            (None,...,...,...)        .......
-----------------------------------------------------------------
activation_4 (Activation)    (None,...,...,...)        .......
-----------------------------------------------------------------
max_pooling2d_4 (MaxPooling2 (None,...,...,...)        .......
-----------------------------------------------------------------
flatten_1 (Flatten)          (None,...)                .......
-----------------------------------------------------------------
dense_1 (Dense)              (None,...)                .......
-----------------------------------------------------------------
batch_normalization_1 (Batch (None,...)                .......
-----------------------------------------------------------------
activation_5 (Activation)    (None,...)                .......
-----------------------------------------------------------------
dense_2 (Dense)              (None,...)                .......
-----------------------------------------------------------------
activation_6 (Activation)    (None,...)                .......
-----------------------------------------------------------------
dense_3 (Dense)              (None,...)                .......
-----------------------------------------------------------------
activation_7 (Activation)    (None,...)                .......
=================================================================
Total params: ......................
-----------------------------------------------------------------
```

Exercise 4 (2 + 2 + 2 = 6 points):

Text is a challenging domain where several challenges need to be faced to learn successful models. With reference to machine learning on text data answer the following questions:

1. Text requires to be encoded, describe what is *word embedding*, what it is used for, and how it is obtained with the word2vec model.
2. Text comes in sequences, describe the Long Short-Term Memory cell and the architectures which can be used when inference is done online, i.e., one word at the time, and batch, i.e., when the entire sentence is available.
3. Long Short-Term Memory cells solve the vanishing/exploding gradient problem of Recurrent Neural Networks. What is this problem due to? How do they solve it?