



BMI 6015, Fall 2018

Quiz: Neural Network and Keras Basics

Release Date: 11-12-2018

Duration: 30 minutes

Marks: 100 points

A. Part I (60 points): Answer the following questions:

(a) From the machine learning perspective, describe one disadvantage and one advantage of feature scaling. (7 points)

(b) If you build your model using a neural network classifier and the classifier training stops before the maximum number of iterations that you set, explain one situation to show the optimizer convergence and another for divergence. (8 points)

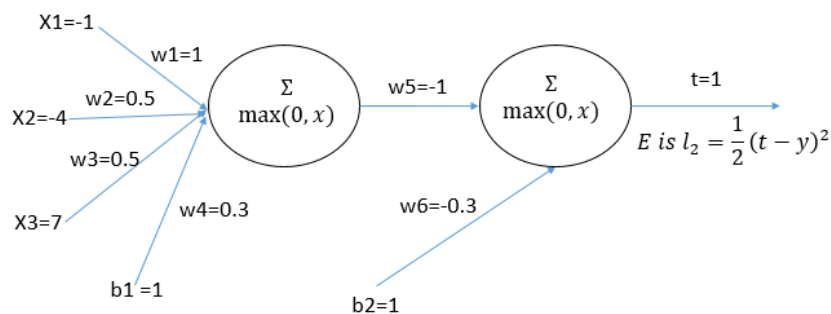
(c) Use the below scaling function to scale each value of $X = (2, -1, 100, 15, 4)$ to be from 1 to 5. Show the calculation of each value. (10 points)

$$f(x) = \frac{(b - a)(x - \min)}{\max - \min} + a$$

(d) If you have 50,000 examples and the batch size equals 10, calculate the number of iterations needed to do 3 epochs over the whole neural network. (5 points)

(e) Use one-hot encoding to encode the categories into 1/0 forms of X= (red, red, blue, green, green, blue). Write the final matrix of the encoded values. (10 points)

(f) Calculate the steps of one feedforward of the neural network shown below: (15 points)



(g) Give an example of a tensor that has 3-dimension of shape = (4, 2, 3). (5 points)

B. Part II (40 points): Use the following Keras code to answer the below questions:

```
#Build the model
from keras import models
from keras import layers
network =models.Sequential()
network.add(layers.Dense(1024,activation='relu',input_shape=(28*28,)))
network.add(layers.Dense(512,activation='sigmoid'))
network.add(layers.Dense(10,activation='softmax'))
network.compile(optimizer='rmsprop',loss='categorical_crossentropy',metrics=['accuracy'])
```

```
network.summary()
```

Layer (type)	Output Shape	Param #
dense_4 (Dense)	(None, 1024)	803840
dense_5 (Dense)	(None, 512)	524800
dense_6 (Dense)	(None, 10)	5130

=====
Total params: 1,333,770
Trainable params: 1,333,770
Non-trainable params: 0

```
network.fit(train_images,train_labels,epochs=5,batch_size=100)
```

```
Epoch 1/5
60000/60000 [=====] - 13s 214us/step - loss: 0.2260 - acc: 0.9313
Epoch 2/5
60000/60000 [=====] - 12s 208us/step - loss: 0.0818 - acc: 0.9747
Epoch 3/5
60000/60000 [=====] - 12s 205us/step - loss: 0.0552 - acc: 0.9836
Epoch 4/5
60000/60000 [=====] - 12s 207us/step - loss: 0.0401 - acc: 0.9874
Epoch 5/5
60000/60000 [=====] - 12s 205us/step - loss: 0.0292 - acc: 0.9910
```

```
test_loss, test_acc = network.evaluate(test_images, test_labels)
print('test_acc', test_acc)

#Overfitting?

10000/10000 [=====] - 1s 107us/step
test_acc 0.9802
```

Questions:

B.1) How many hidden layers are in the network? For each network layer, show how to calculate the number of parameters. (20 points)

B.2) What is the type of task that the network solves (regression, binary classification, or multi-class classification) and why? (5 points)

B.3) Is the network stable over epochs? Why? Is there any overfitting problem? Why? (15 points)