1072 Deep Learning – Homework 2

Due: May 24, 2019, 11:55pm

1. (50%) Please use logistic regression and neural net to classify two datasets, “Iris” and “Ionosphere.’ You can check UCI repository to see more details regarding these datasets.

Iris - <http://archive.ics.uci.edu/ml/datasets/Iris> (15%)

Ionosphere - <https://archive.ics.uci.edu/ml/datasets/ionosphere>

To save you some time, you could use “fisheriris.mat” and “ionosphere.mat” released to you.

The “Iris” dataset contains 150 feature vectors in 4 dimensions, where the label contains three different values, {setosa, versicolor, virginica}, each of which refers to a type of iris plant.

The “Ionosphere” dataset contains radar data collected by a system in Goose Bay, Labrador, which consists of a phased array of 16 high-frequency antennas with a total transmitted power on the order of 6.4 kilowatts. It has 34 real-valued features and categorical response: "b" for bad radar returns and "g" for good radar returns.

* 1. First randomly select 80% of the dataset as your training set and the rest 20% as your testing set. (5% for each)

這邊我就用剛好 ML 講到的 scikit-learn 的 function: “train\_test\_split”

irisSpecies\_train, irisSpecies\_test, irisMeas\_train, irisMeas\_test =

train\_test\_split(irisSpecies, irisMeas, test\_size=0.2)

ionosphereX\_train, ionosphereX\_test, ionosphereY\_train, ionosphereY\_test =

train\_test\_split(ionosphereX, ionosphereY, test\_size=0.2)

* 1. Train logistic regression and neural net classifiers for these two datasets and report accuracy for training and testing datasets as (20% for each)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Classifier | Iris | | Ionosphere | |
| Training | Testing | Training | Testing |
| Logistic Regression | 0.96667 | 0.96667 | | 0.925000 | 0.887324 | |
| Neural Net | 0.975 | 1.0 | | 0.98928 | 0.91549 | |

Please detail your models in the report. For neural net, your model must at least have three hidden layers, meaning including the input and output layers, in total, it has at least five layers. You are encouraged to try to use more layers or neurons in your model to see if they help increase the prediction accuracy.

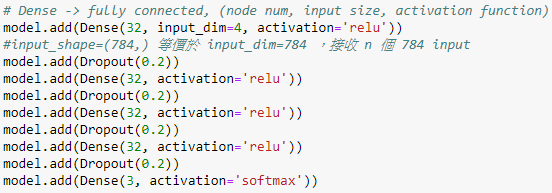
Logistic regression 的部分都是使用 scikit learn:

正在插入影像...

solver 選用 Limited-memory BFGS，宣告多變數，並將跌代次數提高來確保收斂。

Neural Net: 使用 Keras

Iris 的部分

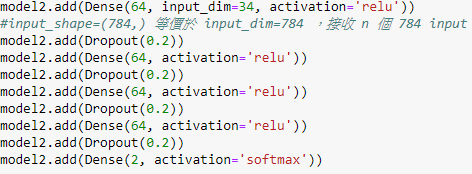


除了最後輸出想要分類外，其它每層都照老師說最常用的 ReLU，並每層都 dropout 20% 避免 overfitting。



因為是多個分類，所以用 categorical\_crossentropy，optimizer 用老師上課提到最常見的 Adam 。

Ionosphere:



因為 input 較多，就想說 node 都改 64 個。因為 binary classifi 輸出層改 2。



用 binary\_crossentropy 當 loss，一樣用 adam 當 optimizer

詳情可以看 Question1.ipynb

1. (30%) In the mnist example where a classification model based on convolutional neural networks is provided, please add 10%, 20%, and 40% of noise to all the images and report the prediction accuracy for the training and testing datasets.

To add noise to the images, you could use the example code below:

import random

import numpy as np

noise\_lv = 0.1

img\_size = 28\*28

for i in range(len(X\_train)):

ran\_seq = random.sample([n for n in range(img\_size)], np.int(img\_size\*noise\_lv))

x = X\_train[i].reshape(-1, img\_size)

x[0, ran\_seq]=255

==10%==

Train: 0.9897

Test: 0.9843

==20%==

Train: 0.9897

Test: 0.9827

==40%==

Train: 0.9818

Test: 0.9703

詳情可看修改過 mnist.ipynb

1. (20%) Following the previous question, please design a noise removal model based on convolutional neural networks for the images with added noise. Report the mean absolute error for the training and testing datasets.

參考了 <https://arxiv.org/pdf/1710.04026.pdf> 的架構，加入了 BatchNormalization ，沒加之前的結果 0 的圖直接變成全黑，數字跟雜訊都一起被濾掉了，而且帶入判斷模型的結果 10 % 多，跟猜的差不多



對於 40% 雜訊的結果:

== Noise remove accuracy ==

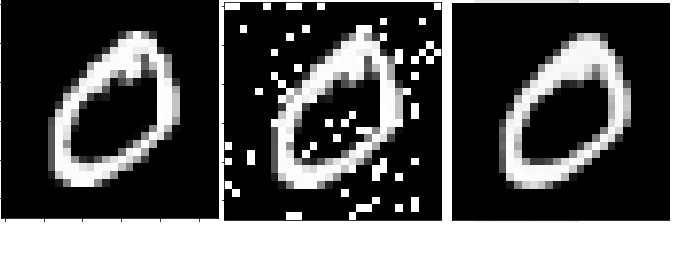
Train loss: 0.016546125790973504

Train accuracy: 0.8143690053939819

Test loss: 0.01638315934240818

Test accuracy: 0.8134149232864379

圖片的結果: (原圖, 雜訊, 移除雜訊)



詳情可看 Question3.ipynb

1. (10% Bonus) Please use the noise removal model to pre-process the images before training the classification model. Would this help improve the prediction accuracy for the images with 10%, 20%, and 40% of noise?

都是用 40% 雜訊移除的 model 來濾掉雜訊，因為我想說 40% 這樣抗雜訊能力最強

10%: 0.986

20%: 0.9881

40%: 0.9854

因為每次 random sample 出來的東西不太一樣，所以每次跑都會有些差異。但原本雜訊的 model 很明顯能看出來準確度是有明顯下降的。而在經過移除雜訊 model 後，效果十分顯著，三者的準確度變得很接近。

詳情可看 Question3.ipynb 的最後面