soln_hw0

May 5, 2022

0.1 HW0 Solutions

0.1.1 1.1 XOR Function

[1]:
```
# Import necessary libraries
import os
import pathlib
import collections
import numpy as np
from pprint import pprint
from tqdm.notebook import tqdm
from nltk.tokenize import word_tokenize
from sklearn.metrics import accuracy_score
from sklearn.metrics import f1_score
from sklearn.neural_network import MLPClassifier
from sklearn.datasets import make_classification
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report
from sklearn.model_selection import train_test_split
from nltk.tokenize.treebank import TreebankWordDetokenizer
```

[2]:
```
# Prepare training data
X = [[0, 0], [0, 1], [1, 0], [1, 1]]
y = [-1, 1, 1, -1]

X = np.array(X)
y = np.array(y)
```

[3]:
```
# Define Neural Network Architecture
nn = MLPClassifier(random_state=1, max_iter=1000, hidden_layer_sizes=(10,),␣
 ↪activation='relu', alpha=0.001, solver='lbfgs', verbose = True)
```

[4]:
```
# Train the model
trained_nn = nn.fit(X, y)
```

[5]:
```
# Test the model
trained_nn.predict([[0,0], [0, 1], [1, 0], [1, 1]])
```

[5]:
```
array([-1,  1,  1, -1])
```

Below are the required weights and offsets of the trained Neural Network.

[6]:
```
# Print weights
print("Wrights: \n{}".format(trained_nn.coefs_))
print()
print("Offsets: \n{}".format(trained_nn.intercepts_))
```

```
Wrights:
[array([[ 1.33447722,  0.88993475,  0.0718904 , -2.72737639, -1.5009656 ,
          2.34825282,  0.04512008,  0.02221048,  0.01484625,  0.04568833],
        [-1.33434529,  0.94641879,  0.04250382,  2.72737654, -1.50097162,
         -2.3482508 ,  0.0118927 , -0.0084404 ,  0.05171728,  0.02863377]]),
array([[ 1.98915953],
        [-1.74930384],
        [-0.0687789 ],
        [ 4.0950882 ],
        [-2.28751604],
        [ 3.65647067],
        [-0.02801403],
        [-0.05026362],
        [ 0.0723572 ],
        [ 0.00704525]])]

Offsets:
[array([-1.31317199e-04,  1.48342393e+00, -2.63858058e-01,  3.60951478e-07,
         1.50097292e+00, -1.09626017e-06, -5.86836104e-01, -6.51874977e-01,
        -4.66930299e-01,  1.12471797e-01]), array([-0.4665509])]
```

0.2 Grading Scheme

- For building training set: 1 point
- For modelling the Neural Networks (with one hidden layer and RelU activation): 1.5 point
- For training the Neural Network: 1.5 point
- For printing the weights and offsets: 2 points
  - One gets full credit for printing **only** *weights* or **only** *offsets*

0.2.1 1.2 Text Classification

[7]:
```
pos_reviews_path = pathlib.Path('A0/review_polarity/txt_sentoken/pos')
neg_reviews_path = pathlib.Path('A0/review_polarity/txt_sentoken/neg')
```

[8]:
```
pos_reviews_files = [name for name in os.listdir(pos_reviews_path) if name[0]!
  ↪='.']
neg_reviews_files = [name for name in os.listdir(neg_reviews_path) if name[0]!
  ↪='.']
```

[9]:
```
reviews_data = []

for name in tqdm(pos_reviews_files):
```

```
    with open(os.path.join(pos_reviews_path, name), 'r') as f_in:
        data = f_in.read()
        tokenized_data = word_tokenize(data)

        reviews_data.append([tokenized_data, 1])


for name in tqdm(neg_reviews_files):
    with open(os.path.join(neg_reviews_path, name), 'r') as f_in:
        data = f_in.read()
        tokenized_data = word_tokenize(data)

        reviews_data.append([tokenized_data, 0])
```

```
  0%|          | 0/1000 [00:00<?, ?it/s]
```

```
  0%|          | 0/1000 [00:00<?, ?it/s]
```

[10]:
```
X_data, y_data = zip(*reviews_data)

X_train, X_test, y_train, y_test = train_test_split(X_data, y_data, test_size=0.
↪2, random_state=123)
```

[11]:
```
pos_lexicon_path = pathlib.Path('A0/opinion_lexicon_English/positive-words.txt')
neg_lexicon_path = pathlib.Path('A0/opinion_lexicon_English/negative-words.txt')
```

[12]:
```
with open(pos_lexicon_path, 'r') as f_in:
    data = f_in.read()
    data = data.split(';;\n\n')
    data = data[-1]
    pos_lexicon_list = word_tokenize(data)
```

[13]:
```
with open(neg_lexicon_path, 'r') as f_in:
    data = f_in.read()
    data = data.split(';;\n\n')
    data = data[-1]
    neg_lexicon_list = word_tokenize(data)
```

0.3 Grading Scheme

- For reading the files (lexicon files and review files): 1 point
- For splitting the files randomly (400 Test review and 1600 Train review): 1 point

1.2.1 Sentiment lexicon-based classifier

[14]:
```
y_pred_lexi_classifier = []

for X, y in tqdm(list(zip(X_test, y_test))):
    pos_lexicon_counts = sum([X.count(l) for l in pos_lexicon_list])
```

```
    neg_lexicon_counts = sum([X.count(l) for l in neg_lexicon_list])

    if pos_lexicon_counts>=neg_lexicon_counts:
        y_pred_lexi_classifier.append(1)
    else:
        y_pred_lexi_classifier.append(0)
```

```
  0%|          | 0/400 [00:00<?, ?it/s]
```

[15]:
```
accuracy_lexi_classifier = accuracy_score(y_test, y_pred_lexi_classifier)
f1_score_lexi_classifier = f1_score(y_test, y_pred_lexi_classifier,␣
 ↪average='binary')
```

[16]:
```
print("Accuracy: {}\nBinary F1 Score: {}".format(accuracy_lexi_classifier,␣
 ↪f1_score_lexi_classifier))
```

```
Accuracy: 0.6825
Binary F1 Score: 0.6558265582655827
```

0.4 Grading Scheme

- For building the classifier: 2.5 point
- For reporting the metrics: 1.5 point
  - One will get full credit for reporting *micro-F1* or *macro-F1* too.

1.2.2 Logistic Regression Classifier

[17]:
```
#Features (no. pos lexicons, no. neg lexicons)

X_train_logistic = [(sum([x.count(l) for l in pos_lexicon_list]), sum([x.
 ↪count(l) for l in neg_lexicon_list])) for x in tqdm(X_train)]
X_test_logistic = [(sum([x.count(l) for l in pos_lexicon_list]), sum([x.
 ↪count(l) for l in neg_lexicon_list])) for x in tqdm(X_test)]
```

```
  0%|          | 0/1600 [00:00<?, ?it/s]
```

```
  0%|          | 0/400 [00:00<?, ?it/s]
```

[18]:
```
regressor = LogisticRegression(random_state=123)
trained_regressor = regressor.fit(X_train_logistic, y_train)
```

[19]:
```
y_pred_logi_classifier = trained_regressor.predict(X_test_logistic)
```

[20]:
```
accuracy_logi_classifier = accuracy_score(y_test, y_pred_logi_classifier)
f1_score_logi_classifier = f1_score(y_test, y_pred_logi_classifier)
```

[21]:
```
print("Accuracy: {}\nBinary F1 Score: {}".format(accuracy_logi_classifier,␣
 ↪f1_score_logi_classifier))
```

```
Accuracy: 0.6725
Binary F1 Score: 0.6666666666666667
```

0.5 Grading Scheme

- For extracting features: 1 point
  - Feature selection is subjective. One will get full credit for selecting any set of *valid* features.
- For building the classifier: 2 point
- For reporting the metrics: 1 point
  - One will get full credit for reporting *micro-F1* or *macro-F1* too.

Note: - One gets 1 point for evaluating both the classifiers on the same test set. - One gets this 1 point trivially for building only one classifier.

Analysis

[22]:
```
# I. Actual: pos; Lex Classifier: pos; Reg Classifier: neg
for x, y1, y2, y3 in zip(X_test, y_test, y_pred_lexi_classifier,␣
 ↪y_pred_logi_classifier):
    if y1==1 and y2==1 and y3==0:
        detokenized_x = TreebankWordDetokenizer().detokenize(x)
        print("Required Review:\n\n{}".format(detokenized_x))
        break
```

```
Required Review:

it's a curious thing - i've found that when willis is not called on to carry the
whole movie, he's much better and so is the movie . even though, in the sixth
sense he is the "name ", he doesn't have the pivotal role . that honour goes to
haley osment who plays cole sear (cute pun, seer) a 9 year old boy who can see
ghosts . if osment was cute or precious, the director going for the maudlin,
this would be nothing more than a movie-of-the-week, thankfully, osment is not
only better than that, but in some instances, blows everyone else off the screen
in a bravura performance . we get to see his fears, vulnerabilities, strengths
and intelligence which makes the sixth sense one of the best movies i've seen
this year . the whole cast matches him in quality, with willis giving a fairly
low key performance that matches the subject matter . one thing about this
movie, its target . this isn't a sfxfest like the haunting or a gorefest, this
is more what i'd call a supernatural drama, more interested in characters than
in dazzling you with makeup . one caveat: there's a lovely twist in the movie,
something like the usual suspects, where you end up replaying the movie in your
head rethinking what you have just seen . i was extremely lucky to see it as a
sneak preview in toronto, before any hype or critical reviews were out, so i
went in with no biases . if anyone want to talk to about the movie before you
see it, don't let them . let the director explain on his own pace and you'll
enjoy the movie vastly more.
```

[23]:
```
# II. Actual: pos; Lex Classifier: neg; Reg Classifier: pos
for x, y1, y2, y3 in zip(X_test, y_test, y_pred_lexi_classifier,␣
 ↪y_pred_logi_classifier):
    if y1==1 and y2==0 and y3==1:
```

```
        detokenized_x = TreebankWordDetokenizer().detokenize(x)
        print("Required Review:\n\n{}".format(detokenized_x))
        break
```

Required Review:

the 1990s produced two brilliant science fiction films . one was _gattaca_ . the other was _the thirteenth floor_ . just as _gattaca_ was overshadowed by the mighty _titanic_, _the thirteenth floor_ was relegated to obscurity by _the matrix_ . however, _the thirteeth floor_, though it deals with similar themes, is a much better movie than the frenetic, childish and improbable _matrix_ . a cutting edge computer scientist, hannon fuller (played by the charming armin mueller-stahl), is murdered . his associate, douglas hall, (craig bierko) apparently framed and suspected by the police, enters into the simulated world they have created in order to unravel the mystery . along the way a beautiful blonde (gretchen mol), more bodies, and a deepening mystery about simulated worlds complicate the picture . _the thirteenth floor_ unfolds slowly and telegraphs its punches, choosing to elicit the more complex emotional response of empathy and anticipation rather than the cheap one of mere surprise . the result is a movie that is a failure from two conventional points of view . first, its plot revelations can be foreseen if one has carefully followed its complex storyline . second and more seriously, it demands that its audience think and feel . no wonder it fell through the cracks . done in film noir style as a murder mystery, this is a relatively deep movie that functions on several levels . many audiences will simply be bewildered by it, rather than engaged, which is a shame, because this movie amply repays a little emotional and intellectual investment . i recommend a second viewing simply to get the flavor of the frequent ironic foreshadowing in the opening parts of the movie . despite its philosophical challenges, _the thirteenth floor_ derives its emotional force from the love story between jane fuller (mol) and douglas hall (bierko), as well as the close friendship between douglas hall and hannon fuller . in that sense it is more akin to _gattaca_ than _the matrix_, which, for all of its cute philosophical byplay, is an adolescent movie with no emotional depth whatsoever . the moment when jane fuller confesses her love for douglas hall is at once satisfying, wrenching and intellectually challenging . in its final moments the film even takes on itself, when david, jane's husband, who has come to enjoy killing, accuses her of being the sick one . although there is a victory for the leading characters, this resolution suggests a disturbing element of fantasy in jane, and thus a deep-seated character flaw . in fact, hannon fuller's activities in the simulated world, the behavior of jane's husband, and the effects of entering the simulation on douglas hall all hint at similar issues with those characters . this is not a film of good people beset from without by great evil . the tragedy is not in their stars, but in themselves . the major actors, all of whom are required to play two or even three roles, perform extremely well . bierko, who will be familiar as the psycho from _the long kiss goodnight_, is outstanding . d'onofrio as whitney, who plays a pivotal role in support, also turns in a good performance . the lovely gretchen mol, whose elegance, integrity, and determination suggest bergman in _casablanca_, does a

wonderful job . her voice, however, lacks the necessary weight at times . this could be because she was given the worst lines in a movie whose major weakness is the script . some minor flaws, hairline cracks in fine porcelain, appear in places . there are one or two instances of jerky editing . the script takes the edge off the film's climaxes . the atmosphere becomes too claustrophobic at times (yet, there is a clue there too). one wonders if even after two decades, _blade runner_ is still casting its long shadow over sci-fi films . one can see a group of suits in their suite, waving their hands imperiously at directors like the emperor in _amadeus_: "make it more like_blade runner_, you know, dark and rainy . "_the thirteen floor_ is that rare exception among hollywood movies: a, rich, emotionally satisfying, intelligent sci-fi movie . and yet, ultimately, it proves the suits right . for when good sci-fi is made, where is the sf community turning out in droves to see it? if we don't support great sf, who will?

[24]:
```
# III. Actual: pos; Lex Classifier: neg; Reg Classifier: neg
for x, y1, y2, y3 in zip(X_test, y_test, y_pred_lexi_classifier,␣
 ↪y_pred_logi_classifier):
    if y1==1 and y2==0 and y3==0:
        detokenized_x = TreebankWordDetokenizer().detokenize(x)
        print("Required Review:\n\n{}".format(detokenized_x))
        break
```

Required Review:

part one of "the strangest movies ever made "series at www . mediajunkies . com by scott watson [light spoilers, nothing you shouldn't read] existenz has been called "a gooey matrix ", and while this is a correct description in some ways, in others it is completely off . the only thing the two films have in common is the idea of a virtual reality that is indistinguishable from normal, "real "reality . a major difference is that "the matrix "has a base reality that is obviously the true reality, while existenz does the impossible and makes us accept the most bizarre reality ever as true . and oh what a reality it is . david cronenberg has shown us before that he knows how to create strange movies (naked lunch, crash), but this one is one of the strangest . maybe it's because of how the mundane is mixed with the bizarre . technology is organic, but only some of it . cars are normal . gas stations are the same . cell phones are glowing bug like appendages . you jack into a virtual reality game using a quivering pink gooey animal like apparatus (referred to simply as a "pod "). high technology has been replaced with organic creatures, and it provides a truly surreal setting for the characters . a famous game designer, allegra geller (a fine looking jennifer jason leigh), flees a demonstration of her latest game, existenz, after an assassination attempt . she is on the run (from who, i'm still not sure), and has to save her pod by playing her game with "someone friendly ". that someone friendly is her dubious companion and security guard, ted pikul (jude law). unfortunately he is lacking a fundamental requirement for playing, a bioport . sort of a hole in your spine (ok, exactly like a hole in your spine), a bioport allows you to plug the vaguely entrail-

looking wires from the pod into you, allowing you to play the game . luckily a bioport can be installed easily (they install them in malls), so off they go to get one . after a crazy series of events at the local country gas station, they are able to plug in and play existenz . the plot itself is a surreal melding of conspiracies, reality distortion, chinese restaurants, and multi appendaged amphibian things ("it's a sign of the times "). but the plot, while cohesive in it's own creepy way is secondary to just trying to figure out what the hell is going on in this universe . cronenberg just slaps this weird organic-tech world into the viewers lap, and it's up to you to figure out how all these things work . and that's the true joy of this film, entering the vaguely nightmarish world in which existenz takes place . you learn a little bit piece by piece (oh, they make pods out of those), and it's just enough to get you to accept the world that is presented to you . there are all kinds of twists and turns as the characters wander around, including a fantastic scene in a chinese restaurant that will leave you trying to assemble something out of your moo goo gai pan for months afterwards . also the bioports are occasionally used for things other than porting into existenz, the specifics of which i will leave for you to discover . there's lots of little things that set off the weird-o-meter, like the little tiny pods that you can buy in the video game store, the two headed amphibian creature, and why that guy at the gas station's name appears to be "gas ". it all adds up to a supremely far out trip, and given the choice i think i would rather live in the matrix's virtual world . . . oh wait, i already do.

0.6 Grading Scheme

- For printing each type of reviews: Total 3 points (1 point each)
- One gets full credit for using reviews from the test set, instead of writing their own examples.