Training Binary Classifiers

Thursday, February 27, 2020   4:09 PM

Note Written by: Zulfadli Zainal

Lets simplify to only detecting 1 digit -> Example: T detect Number 9 (Like a 9 detector tools)
This tools is capable to detect just 2 classes -> Whether its 9 or not 9.

To start

Now, create a target vector for binary classifiers.

```
#Try to predict some_digit = 9 using SGD (Stochastic Gradient Descent)
y_train = y_train.astype(np.int8)
y_test = y_test.astype(np.int8)

y_train_9 = (y_train == 9)
y_test_9 = (y_test == 9)
```

```
#Create a Boolean from labelled data.

y_train = y_train.astype(np.int8)
y_test = y_test.astype(np.int8)

y_train_9 = (y_train == 9)
y_test_9 = (y_test == 9)
```

Result:

⊞ y_train_9 - NumPy array

	0
45	False
46	False
47	False
48	True
49	False
50	False
51	True
52	False
53	False
54	True
55	False
56	False
57	False

Format	Resize	☑

Strat with SGD (Stochastic Gradient Descent)

SGD - Good for handling large datasets efficiently. It also deals with training instances independently one at a time. Good for online learning.

Create SGD Classifier:

```python
from sklearn.linear_model import SGDClassifier
```

```python
sgd_clf = SGDClassifier(random_state=42)
sgd_clf.fit(X_train, y_train_9)

#Now can use the SGD to detect the image of number 9
print(sgd_clf.predict([some_digit]))
```

```python
sgd_clf = SGDClassifier(random_state=42)
sgd_clf.fit(X_train, y_train_9)

#Now can use the SGD to detect the image of number 9
print(sgd_clf.predict([some_digit]))
```

Result:

[True] -> It can detect out picture is truly 9!