Classification

Classification

Binary Classification results in a decision that is either true or false.
Binary classification Examples:

- Classify whether a medical case is positive or negative, or whether an image contains a hotdog or not a hotdog.
- Classify whether an email is spam or not spam.

Multi-class classification categorizes examples in one of several potential categories (always three or more).
Multi-class classification Examples:

- Classifying whether air quality is poor, moderate, or severe.
- Classifying text into topics such as sports, entertainment, politics, and so on.

The picture describes a joke app designed by one of the characters in the Silicon Valley show that classifies food items from images. However, when revealed, the app was able to only classify hot dogs in images, but for all other food items, it would just classify them as not hotdog.

Cross-Entropy Loss

Cross-entropy is a score that summarizes the average difference between the actual and predicted probability distributions for all classes. In a classification model, the goal is to minimize the score, with a perfect cross-entropy value is 0. We can calculate cross-entropy loss by using the `log_loss()` function in scikit-learn.

```
# example implementation of cross-entropy loss

true_labels = [1, 0, 0]
predicted_labels = [0.7, 0.2, 0.1]
print(log_loss(true_labels, predicted_labels))
```

codecademy

Preparing Your Data

To prepare data for cross-entropy loss analysis, you can use the `to_categorical()` function in TensorFlow's Keras API to convert labels into one-hot-encodings.

```
updated_y_train =
tensorflow.keras.utils.to_categorical(y_
train, dtype = 'int64')
updated_y_test =
tensorflow.keras.utils.to_categorical(y_
test, dtype = 'int64')
```

Classification Loss

When performing a deep learning classification model, one common `loss` parameter is `categorical_crossentropy`. Another loss parameter one can use for deep learning classification models is `sparse_categorical_crossentropy`, which is a computationally modified categorical cross-entropy loss that allows integer labels to be left as they are to avoid the procedure of encoding.
We can set a model's loss parameter in the Keras API with TensorFlow as depicted in the code block.

```
# categorical cross-entropy
my_model.compile(loss='categorical_cross
entropy', optimizer='adam',
metrics=['accuracy'])

# sparse categorical cross-entropy
model.compile(loss='sparse_categorical_c
rossentropy', optimizer='adam',
metrics=['accuracy'])
```

codecademy

F1-Score

In a deep learning classification model, an *F1-score* can be used to evaluate how our model performs based on how poorly it makes false negative mistakes.

In the code snippet shown, we do the following:

- predict classes for all test cases `my_test` using the scikit-learn `.predict()` method and assign the result to the `yhat_classes` variable.
- convert the one-hot-encoded labels `my_test_labels` into the index of the class the sample belongs to using `.argmax()` from the NumPy library. The index corresponds to our class encoded as an integer.
- use the `.classification_report()` method from the scikit-learn library to calculate all the metrics.

Print Share

```
import numpy as np
from sklearn.metrics import
classification_report

yhat_classes =
np.argmax(my_model.predict(my_test),
axis = -1)
y_true = np.argmax(my_test_labels,
axis=1)
print(classification_report(y_true,
yhat_classes))
```