Classification models

Puteaux, Fall/Winter 2020-2021

```
################################
##                            ##
##  Deep Learning in Python   ##
##                            ##
################################
```

```
§1 Introduction to Deep Learning in Python
```

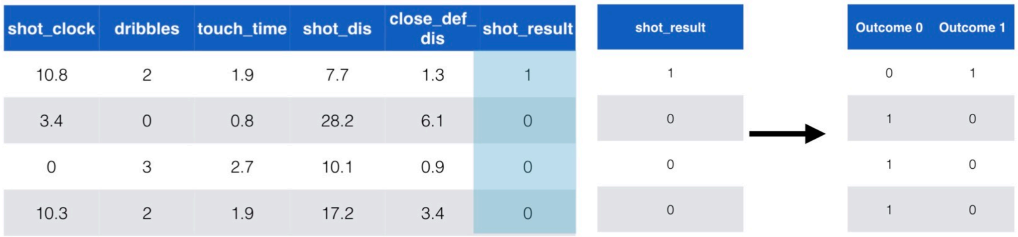
```
§1.3 Building deep learning models with keras
```

```
§1.3.3 Classification models
```

1. How to compile the classification model with Keras?

- Use `'categorical_crossentropy'` loss function, which is similar to log loss, but lower is better.
- Add `metrics = ['accuracy']` to compile step for easy-to-understand diagnostics.
- The output layer has a separate node for each possible outcome and uses `'softmax'` activation.

2. How to transform the target value into categorical?



shot_clock	dribbles	touch_time	shot_dis	close_def_dis	shot_result
10.8	2	1.9	7.7	1.3	1
3.4	0	0.8	28.2	6.1	0
0	3	2.7	10.1	0.9	0
10.3	2	1.9	17.2	3.4	0

shot_result
1
0
0
0

Outcome 0	Outcome 1
0	1
1	0
1	0
1	0

3. Code of classification:

[1]:

```
import pandas as pd
from keras.layers import Dense
from keras.models import Sequential


def Data_preparation(df):
    df = df.reindex(columns=[
        'SHOT_CLOCK', 'DRIBBLES', 'TOUCH_TIME', 'SHOT_DIST', 'CLOSE_DEF_DIST',
        'SHOT_RESULT'
    ])
    df['SHOT_CLOCK'] = df['SHOT_CLOCK'].fillna(0)
    df['SHOT_RESULT'].replace('missed', 0, inplace=True)
    df['SHOT_RESULT'].replace('made', 1, inplace=True)
    df.columns = df.columns.str.lower()
    return df
```

[2]:

```
from keras.utils.np_utils import to_categorical

data = pd.read_csv('ref5. Basketball shot log.csv')

data = Data_preparation(data)

predictors = data.drop(['shot_result'], axis=1).to_numpy()
n_cols = predictors.shape[1]
target = to_categorical(data.shot_result)

model = Sequential()
model.add(Dense(100, activation='relu', input_shape=(n_cols, )))
model.add(Dense(100, activation='relu'))
model.add(Dense(100, activation='relu'))
model.add(Dense(2, activation='softmax'))
model.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])

model.fit(predictors, target)
```

```
4003/4003 [==============================] - 8s 2ms/step - loss: 0.6624 -
accuracy: 0.6064
```

[2]:

```
<tensorflow.python.keras.callbacks.History at 0x7fdaee483050>
```

4. Practice question for understanding the classification data:

- To start modeling with a new dataset for a classification problem. This data includes information about passengers on the Titanic. The predictors such as `age`, `fare`, and where each passenger embarked to could be used to predict who will survive. This data is from a tutorial on data science competitions. There are descriptions of the features.

- It's smart to review the maximum and minimum values of each variable to ensure the data isn't misformatted or corrupted. What was the maximum age of passengers on the Titanic? Use the `.describe()` method in the IPython Shell to answer this question.

☐ 29.699.

☒ 80.

□ 891.

□ It is not listed.

▶ **Package pre-loading:**

[3]:
```
import pandas as pd
```

▶ **Data pre-loading:**

[4]:
```
df = pd.read_csv('ref6. Titanic.csv')
```

▶ **Question-solving method:**

[5]:
```
df.head()
```

[5]:
```
   survived  pclass   age  sibsp  parch     fare  male  age_was_missing  \
0         0       3  22.0      1      0   7.2500     1            False
1         1       1  38.0      1      0  71.2833     0            False
2         1       3  26.0      0      0   7.9250     0            False
3         1       1  35.0      1      0  53.1000     0            False
4         0       3  35.0      0      0   8.0500     1            False

   embarked_from_cherbourg  embarked_from_queenstown  \
0                        0                         0
1                        1                         0
2                        0                         0
3                        0                         0
4                        0                         0

   embarked_from_southampton
0                          1
1                          0
2                          1
3                          1
4                          1
```

[6]:
```
df['age'].describe()
```

[6]:
```
count    891.000000
mean      29.699118
std       13.002015
min        0.420000
25%       22.000000
50%       29.699118
75%       35.000000
max       80.000000
Name: age, dtype: float64
```

[7]:

```python
max_age = int(df['age'].max())
print('The maximum age of passengers on the Titanic is {}.'.format(max_age))
```

```
The maximum age of passengers on the Titanic is 80.
```

5. Practice exercises for classification models:

▶ Package pre-loading:

[8]:

```python
import pandas as pd
```

▶ Data pre-loading:

[9]:

```python
df = pd.read_csv('ref6. Titanic.csv')

df['age_was_missing'].replace(False, 0, inplace=True)
df['age_was_missing'].replace(True, 1, inplace=True)

predictors = df.drop(['survived'], axis=1).to_numpy()
n_cols = predictors.shape[1]
```

▶ Classification models practice:

[10]:

```python
# Import necessary modules
import keras
from keras.layers import Dense
from keras.models import Sequential
from keras.utils import to_categorical

# Convert the target to categorical: target
target = to_categorical(df.survived)

# Set up the model
model = Sequential()

# Add the first layer
model.add(Dense(32, activation='relu', input_shape=(n_cols, )))

# Add the output layer
model.add(Dense(2, activation='softmax'))

# Compile the model
model.compile(optimizer='sgd',
              loss='categorical_crossentropy',
              metrics=['accuracy'])

# Fit the model
model.fit(predictors, target)
```

```
28/28 [==============================] - 0s 16ms/step - loss: 5.3186 - accuracy:
0.5823
```

```
[10]: <tensorflow.python.keras.callbacks.History at 0x7fdaf0371290>
```