Binary Classification

將建立一個模型，使用二元分類器(Binary Classification)來預測 *Hotel Cancellations*。

設定環境

[1]:
```
# Setup plotting
import matplotlib.pyplot as plt
plt.style.use('seaborn-whitegrid')
# Set Matplotlib defaults
plt.rc('figure', autolayout=True)
plt.rc('axes', labelweight='bold', labelsize='large',
       titleweight='bold', titlesize=18, titlepad=10)
plt.rc('animation', html='html5')

# Setup feedback system
from learntools.core import binder
binder.bind(globals())
from learntools.deep_learning_intro.ex6 import *
```

首先，載入 *Hotel Cancellations* 資料集。

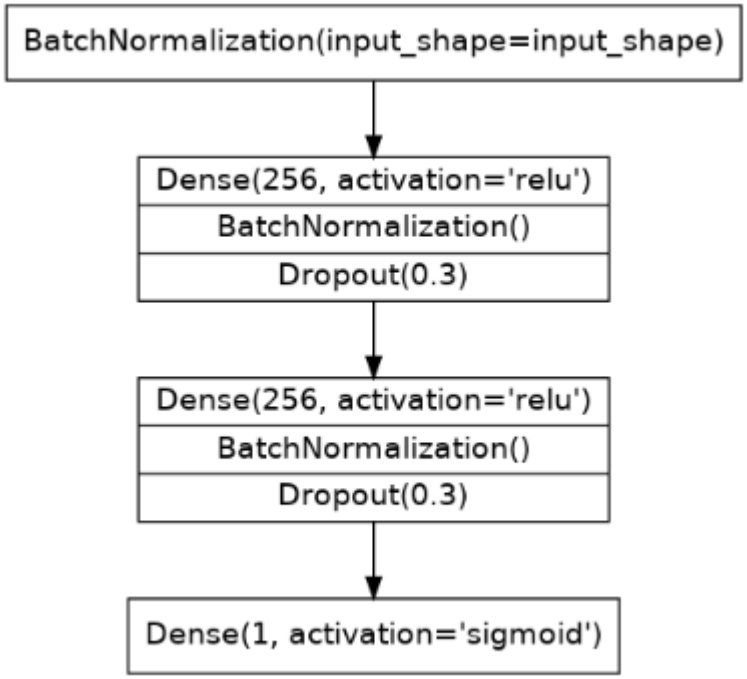
```
import pandas as pd

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.impute import SimpleImputer
from sklearn.pipeline import make_pipeline
from sklearn.compose import make_column_transformer

hotel = pd.read_csv('../input/dl-course-data/hotel.csv')

X = hotel.copy()
y = X.pop('is_canceled')

X['arrival_date_month'] = \
    X['arrival_date_month'].map(
        {'January':1, 'February': 2, 'March':3,
         'April':4, 'May':5, 'June':6, 'July':7,
         'August':8, 'September':9, 'October':10,
         'November':11, 'December':12}
    )

features_num = [
    "lead_time", "arrival_date_week_number",
    "arrival_date_day_of_month", "stays_in_weekend_nights",
    "stays_in_week_nights", "adults", "children", "babies",
    "is_repeated_guest", "previous_cancellations",
    "previous_bookings_not_canceled", "required_car_parking_spaces",
    "total_of_special_requests", "adr",
]
features_cat = [
    "hotel", "arrival_date_month", "meal",
    "market_segment", "distribution_channel",
    "reserved_room_type", "deposit_type", "customer_type",
]
```

```
transformer_num = make_pipeline(
    SimpleImputer(strategy="constant"), # there are a few missing values
    StandardScaler(),
)
transformer_cat = make_pipeline(
    SimpleImputer(strategy="constant", fill_value="NA"),
    OneHotEncoder(handle_unknown='ignore'),
)

preprocessor = make_column_transformer(
    (transformer_num, features_num),
    (transformer_cat, features_cat),
)

# stratify - make sure classes are evenlly represented across splits
X_train, X_valid, y_train, y_valid = \
    train_test_split(X, y, stratify=y, train_size=0.75)

X_train = preprocessor.fit_transform(X_train)
X_valid = preprocessor.transform(X_valid)

input_shape = [X_train.shape[1]]
```

1) Define Model

我們這次使用的模型將包含批量歸一化層和 dropout 層。為了方便閱讀，我們將圖表拆分成多個區塊，但您可以像往常一樣逐層定義。

定義一個具有如下圖架構的模型：




```
from tensorflow import keras
from tensorflow.keras import layers

model = keras.Sequential([
    layers.BatchNormalization(input_shape=input_shape),
    layers.Dense(256, activation='relu'),
    layers.BatchNormalization(),
    layers.Dropout(0.3),
    layers.Dense(256, activation='relu'),
    layers.BatchNormalization(),
    layers.Dropout(0.3),
    layers.Dense(1, activation='sigmoid'),
])
q_1.assert_check_passed()
# Check your answer
q_1.check()
```

2）新增優化器、損失和指標(Add Optimizer, Loss, and Metric)

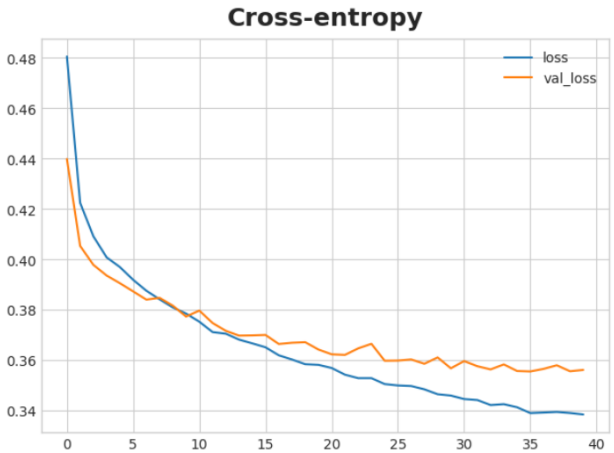
現在使用 Adam 優化器(Adam optimizer)和交叉熵損失(cross-entropy loss)和準確度指標的二進位版本來編譯模型。

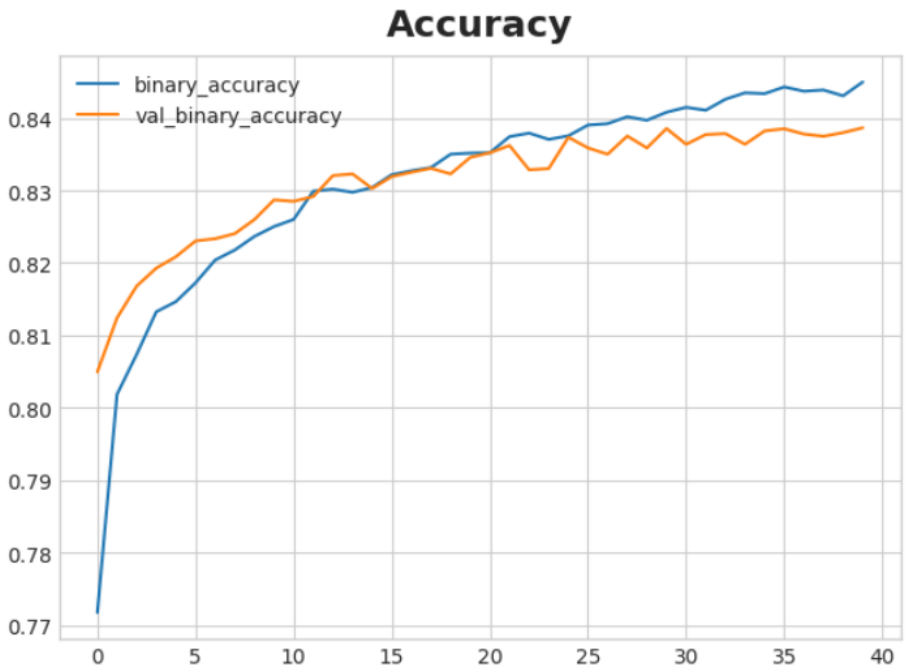
```
model.compile(
    optimizer='adam',
    loss='binary_crossentropy',
    metrics=['binary_accuracy'],
)
```

運行此 cell 來訓練模型並查看學習曲線。它可能需要運行大約 60 到 70 個 epoch，這可能需要一兩分鐘。

```
early_stopping = keras.callbacks.EarlyStopping(
    patience=5,
    min_delta=0.001,
    restore_best_weights=True,
)
history = model.fit(
    X_train, y_train,
    validation_data=(X_valid, y_valid),
    batch_size=512,
    epochs=200,
    callbacks=[early_stopping],
)

history_df = pd.DataFrame(history.history)
history_df.loc[:, ['loss', 'val_loss']].plot(title="Cross-entropy")
history_df.loc[:, ['binary_accuracy', 'val_binary_accuracy']].plot(title="Accuracy")
```

```
175/175 [==============================] - 1s 5ms/step - loss: 0.3391 - binary_accuracy: 0.8437 - val_loss: 0.3565 - val_binary_accuracy: 0.8378
Epoch 38/200
175/175 [==============================] - 1s 5ms/step - loss: 0.3394 - binary_accuracy: 0.8439 - val_loss: 0.3579 - val_binary_accuracy: 0.8375
Epoch 39/200
175/175 [==============================] - 1s 5ms/step - loss: 0.3390 - binary_accuracy: 0.8431 - val_loss: 0.3555 - val_binary_accuracy: 0.8380
Epoch 40/200
175/175 [==============================] - 1s 5ms/step - loss: 0.3384 - binary_accuracy: 0.8450 - val_loss: 0.3561 - val_binary_accuracy: 0.8386
<Axes: title={'center': 'Accuracy'}>
```





3）訓練和評估(Train and Evaluate)

學習曲線怎麼樣？模型看起來是欠擬合(underfit)還是過度擬合(overfit)？交叉熵損失(cross-entropy loss)能很好地取代準確率嗎？

雖然我們可以看到訓練損失持續下降，但提前停止回調阻止了任何過度擬合。此外，準確率的上升速度與交叉熵(cross-entropy)的下降速度相同，因此最小化交叉熵似乎是不錯的替代方案。總而言之，這次訓練看起來是成功的！