EN 685.621 HW3 - Noboru Hayashi

Q1: The script is using a train & test dataset from Kaggle.com. The accuracy reaches 98.775%

```
import pandas as pd
import numpy as np
import tensorflow as tf
from tensorflow.keras import layers, models
import matplotlib.pyplot as plt

if __name__ == '__main__':

   df_train = pd.read_csv('train.csv')
   labels_train = df_train['label']
   images_train = df_train.iloc[:,1:]/255.0
   images_train = images_train.to_numpy().reshape(df_train.shape[0], 28, 28, 1)


   model = models.Sequential()
   model.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(28, 28, 1)))
   model.add(layers.MaxPooling2D((2, 2)))
   model.add(layers.Conv2D(64, (3, 3), activation='relu'))
   model.add(layers.MaxPooling2D((2, 2)))
   model.add(layers.Conv2D(64, (3, 3), activation='relu'))

   model.add(layers.Flatten())
   model.add(layers.Dense(64, activation='relu'))
   model.add(layers.Dense(10, activation='softmax'))

   print(model.summary())


   model.compile(optimizer='adam',
               loss='sparse_categorical_crossentropy',
               metrics=['accuracy'])

   model.fit(images_train, labels_train, epochs=5)


   df_test = pd.read_csv('test.csv')
   images_test = df_test.to_numpy().reshape(df_test.shape[0], 28, 28, 1)

   pred_test = np.argmax(model.predict(images_test), axis=-1)
```

```
output = pd.DataFrame(range(1, len(pred_test)+1))
output.columns=['ImageId']
output['Label'] = pred_test
output.to_csv('submission.csv', index=False)
print('Done! submission file is generated')
```

Output:

```
2021-05-03 18:19:32.154767: I tensorflow/core/platform/cpu_feature_guard.cc:143] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2 FMA
2021-05-03 18:19:32.172669: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x7fc2c43d1c80 initialized for platform Host (this does not guarantee that XLA will be used). Devices:
2021-05-03 18:19:32.172696: I tensorflow/compiler/xla/service/service.cc:176]   StreamExecutor device (0): Host, Default Version
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 26, 26, 32)        320
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 13, 13, 32)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 11, 11, 64)        18496
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 5, 5, 64)          0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 3, 3, 64)          36928
_________________________________________________________________
flatten (Flatten)            (None, 576)               0
_________________________________________________________________
dense (Dense)                (None, 64)                36928
_________________________________________________________________
dense_1 (Dense)              (None, 10)                650
=================================================================
Total params: 93,322
Trainable params: 93,322
Non-trainable params: 0
_________________________________________________________________
None
Epoch 1/5
1313/1313 [==============================] - 17s 13ms/step - loss: 0.1834 - accuracy: 0.9439
Epoch 2/5
1313/1313 [==============================] - 17s 13ms/step - loss: 0.0541 - accuracy: 0.9831
Epoch 3/5
1313/1313 [==============================] - 17s 13ms/step - loss: 0.0375 - accuracy: 0.9885
Epoch 4/5
1313/1313 [==============================] - 17s 13ms/step - loss: 0.0310 - accuracy: 0.9900
Epoch 5/5
1313/1313 [==============================] - 19s 15ms/step - loss: 0.0233 - accuracy: 0.9925
Done! submission file is generated
```

Q2:

0: MIN-MAX-SEARCH is called, the player to make action is X. MAX-VALUE is called. Starting from the board state as below:

```
   -----
   x   o
   o
       x
   -----
```

0->1c: MAX-VALUE() searches action space for X (DFS). For each action, call MIN-VALUE(). For simulation, MIN-VALUE() for the board state as below:

board state 1c:

```
   -----
   x   o
   o x
       x
   -----
```

1c->2b & 2c. MIN-VALUE() is called. DFS to search action space for O. And similary, for each action, max-value() is called

board state 2b: game does not end. continue

```
   -----
   x   o
   o o x
       x
   -----
```

board state 2c: o wins, the v will be -1

```
   -----
   x   o
   o x
   o   x
```

2b->3b: X's turn, MAX-VALUE() for each action is called. In figure 3, MAX-VALUE() is called for the board state below:

board state 3b:

```
   -----
   x   o
   o o x
   x   x
   -----
```

3b->4a & 4b: In figure 3, level 3 to 4 put X's to the board. So X's turn, call MAX-VALUE() for actions below:

board state 4a: Continue

```
   -----
   x x o
```

```
   o o x
   x   x
   -----
   board state 4b: X wins, v=1
   -----
   x   o
   o o x
   x x x
   -----
4a->5: O's turn, min-value() is called to the only action below:
   -----
   x x o
   o o x
   x o x
   -----
level 5 -> 4a: The game ties, v = 0 and method goes upwards, v for 4a is 0.
level 4a & 4b -> 3b: max of the values for two action is 1
level 3b -> 2b: level 2 to 3 was for X's turn, the max value is 1. So v for 2b is 1
level 2b & 2c -> 1c: the min of the values for action 2a, 2b is -1.
level 1c->0: Since we only simulate only one DFS path, the max value from 5 possible
actions is unknown. but we can know the value for the board state 1c is -1, so this
action would never be picked by AI.
```

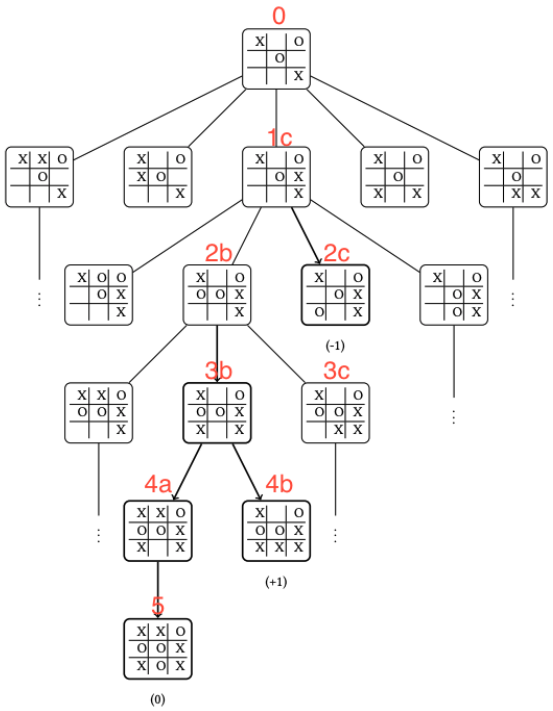

Figure 3: The tic-tac-toe board for use with the search algorithm.