Assignment 5b - Deep Learning

In this assignment, you will

1. Load mnk data and split them into training and test set.
2. Build model based on the given model structure, and implement a custome loss function.
3. Train the model and plot the learning curves with respect to the number of epochs.
4. Evaluate the model on the test set. Report performance metrics.
5. Choose a few game states and print the predictions made by the model on these states.
6. Discuss findings.

Your Information

In [1]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
drive.mount("/content/drive", force_remount=True).
```

In [2]:
```
%cd /content/drive/MyDrive/2022 \files/CS_581
```

```
/content/drive/MyDrive/2022 files/CS_581
```

In [3]:
```
import numpy as np
from numpy.random import RandomState
from collections import Counter
import matplotlib.pyplot as plt
from time import time
import os
import sys
import pickle
from operator import itemgetter
from copy import deepcopy
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, mean_squared_error
import tensorflow as tf
from tensorflow.keras.optimizers import Adam

from utils import MNKNode, ConnectFour, alpha_beta_search
from gn_network import ConvResNN
```

Dataset

Load three files downloaded from BlackBoard.

`mnk_X_data.pickle` represents the game board data.

- Each instance x from mnk_X_data is a (3, 3, 3) vector
- x[0, :, :] records the positions of player O
- x[1, :, :] records the positions of player X
- x[2, 0, 0] represents the next player (0: O, 1: X)

`mnk_y_move.pickle` represents the labels of the next moves for current state.

- There are 9 move labels in total (0, 1, ..., 8), counting from left to right, top to bottom

`mnk_y_value.pickle` represents the outcome labels (0: O wins, 1: X wins, 0.5: draw).

Example of x:

O - O X - X - X -[[[1., 0., 1.], [0., 0., 0.], [0., 0., 0.]], [[0., 0., 0.], [1., 0., 1.], [0., 1., 0.]], [[0., 0., 0.], [0., 0., 0.], [0., 0., 0.]]

Load data from files.

In [4]:
```
X       = pickle.load(open("mnk_X_data.pickle",   "rb"))
y_move  = pickle.load(open("mnk_y_move.pickle",   "rb"))
y_value = pickle.load(open("mnk_y_value.pickle", "rb"))
```

Split the data into train set and test set.

In [5]:
```
# Split data
train_idx,     test_idx, _, _ = train_test_split(np.arange(len(X)), y_value, te
st_size=1./3, stratify=y_value, random_state=1234)
X_train,       X_test         = X[train_idx], X[test_idx]
y_move_train,  y_move_test    = y_move[train_idx], y_move[test_idx]
y_value_train, y_value_test   = y_value[train_idx], y_value[test_idx]
```

Model Fit

Complete the TODO parts in **gn_network.py**. You will write a custom model with following structure by subclassing the `Model` class. See option 2 in https://www.tensorflow.org/api_docs/python/tf/keras/Model (https://www.tensorflow.org/api_docs/python/tf/keras/Model) for reference:

1. Conv2D Layer1 (128 filters, 3*3 kernel, 1*1 strides, 'same' padding, 'relu' activation)
2. Conv2D Layer2 (64 filters, 3*3 kernel, 1*1 strides, 'same' padding, 'relu' activation)
3. Conv2D Layer3 (32 filters, 3*3 kernel, 1*1 strides, 'same' padding, 'relu' activation)
4. Flatten Layer
5. Dense Layer1 (32 units, 'relu' activation)
6. Dense Layer move (9 units, 'softmax' activation)
7. Dense Layer value (1 units, 'sigmoid' activation)

Set Adam with learning rate 1e-3 as the optimizer to compile the model.

Set the number of epochs to 100, validation_split to 0.2, and then fit the model.

In [6]:
```
model = ConvResNN()
optimizer = Adam(1e-3)
model.compile(optimizer)

start_time = time()
history = model.fit(X_train, [y_move_train, y_value_train],
                    epochs=100,
                    validation_split=0.2)
running_time = time() - start_time
```

```
Epoch 1/100
76/76 [==============================] - 4s 15ms/step - train hybrid loss: 1.9
430 - train m accuracy: 0.3869 - train v accuracy: 0.2581 - train m loss: 1.79
13 - train v loss: 0.1517 - val_test hybrid loss: 1.6220 - val_test m accurac
y: 0.5141 - val_test v accuracy: 0.2570 - val_test m loss: 1.5049 - val_test v
loss: 0.1170
Epoch 2/100
76/76 [==============================] - 1s 7ms/step - train hybrid loss: 1.37
82 - train m accuracy: 0.5803 - train v accuracy: 0.2590 - train m loss: 1.255
0 - train v loss: 0.1232 - val_test hybrid loss: 1.1935 - val_test m accuracy:
0.6285 - val_test v accuracy: 0.2570 - val_test m loss: 1.0894 - val_test v lo
ss: 0.1041
Epoch 3/100
76/76 [==============================] - 1s 7ms/step - train hybrid loss: 1.03
96 - train m accuracy: 0.6733 - train v accuracy: 0.2634 - train m loss: 0.923
9 - train v loss: 0.1157 - val_test hybrid loss: 1.0842 - val_test m accuracy:
0.6252 - val_test v accuracy: 0.2570 - val_test m loss: 0.9799 - val_test v lo
ss: 0.1042
Epoch 4/100
76/76 [==============================] - 1s 7ms/step - train hybrid loss: 0.83
23 - train m accuracy: 0.7343 - train v accuracy: 0.2438 - train m loss: 0.722
8 - train v loss: 0.1096 - val_test hybrid loss: 0.8753 - val_test m accuracy:
0.7032 - val_test v accuracy: 0.2570 - val_test m loss: 0.7759 - val_test v lo
ss: 0.0993
Epoch 5/100
76/76 [==============================] - 1s 7ms/step - train hybrid loss: 0.73
54 - train m accuracy: 0.7543 - train v accuracy: 0.2489 - train m loss: 0.627
0 - train v loss: 0.1084 - val_test hybrid loss: 0.7967 - val_test m accuracy:
0.7280 - val_test v accuracy: 0.2570 - val_test m loss: 0.6961 - val_test v lo
ss: 0.1006
Epoch 6/100
76/76 [==============================] - 1s 8ms/step - train hybrid loss: 0.62
42 - train m accuracy: 0.7995 - train v accuracy: 0.2469 - train m loss: 0.521
9 - train v loss: 0.1023 - val_test hybrid loss: 0.7581 - val_test m accuracy:
0.7231 - val_test v accuracy: 0.2570 - val_test m loss: 0.6585 - val_test v lo
ss: 0.0995
Epoch 7/100
76/76 [==============================] - 1s 7ms/step - train hybrid loss: 0.57
92 - train m accuracy: 0.8070 - train v accuracy: 0.2683 - train m loss: 0.481
1 - train v loss: 0.0981 - val_test hybrid loss: 0.6879 - val_test m accuracy:
0.7629 - val_test v accuracy: 0.2570 - val_test m loss: 0.5933 - val_test v lo
ss: 0.0946
Epoch 8/100
76/76 [==============================] - 1s 7ms/step - train hybrid loss: 0.51
85 - train m accuracy: 0.8356 - train v accuracy: 0.2591 - train m loss: 0.426
4 - train v loss: 0.0921 - val_test hybrid loss: 0.7252 - val_test m accuracy:
0.7612 - val_test v accuracy: 0.2570 - val_test m loss: 0.6324 - val_test v lo
ss: 0.0928
Epoch 9/100
76/76 [==============================] - 1s 7ms/step - train hybrid loss: 0.48
94 - train m accuracy: 0.8481 - train v accuracy: 0.2692 - train m loss: 0.401
3 - train v loss: 0.0881 - val_test hybrid loss: 0.6403 - val_test m accuracy:
0.7894 - val_test v accuracy: 0.2570 - val_test m loss: 0.5530 - val_test v lo
ss: 0.0873
Epoch 10/100
76/76 [==============================] - 1s 7ms/step - train hybrid loss: 0.43
77 - train m accuracy: 0.8680 - train v accuracy: 0.2500 - train m loss: 0.354
5 - train v loss: 0.0832 - val_test hybrid loss: 0.6048 - val_test m accuracy:
0.8176 - val_test v accuracy: 0.2570 - val_test m loss: 0.5150 - val_test v lo
ss: 0.0898
```

```
Epoch 11/100
76/76 [==============================] - 1s 7ms/step - train hybrid loss: 0.39
81 - train m accuracy: 0.8726 - train v accuracy: 0.2502 - train m loss: 0.322
6 - train v loss: 0.0755 - val_test hybrid loss: 0.6453 - val_test m accuracy:
0.7894 - val_test v accuracy: 0.2570 - val_test m loss: 0.5574 - val_test v lo
ss: 0.0878
Epoch 12/100
76/76 [==============================] - 1s 7ms/step - train hybrid loss: 0.37
20 - train m accuracy: 0.8858 - train v accuracy: 0.2483 - train m loss: 0.297
5 - train v loss: 0.0745 - val_test hybrid loss: 0.6530 - val_test m accuracy:
0.7745 - val_test v accuracy: 0.2570 - val_test m loss: 0.5723 - val_test v lo
ss: 0.0807
Epoch 13/100
76/76 [==============================] - 1s 7ms/step - train hybrid loss: 0.34
38 - train m accuracy: 0.8950 - train v accuracy: 0.2620 - train m loss: 0.276
3 - train v loss: 0.0675 - val_test hybrid loss: 0.6405 - val_test m accuracy:
0.7944 - val_test v accuracy: 0.2570 - val_test m loss: 0.5666 - val_test v lo
ss: 0.0739
Epoch 14/100
76/76 [==============================] - 1s 8ms/step - train hybrid loss: 0.31
99 - train m accuracy: 0.8987 - train v accuracy: 0.2566 - train m loss: 0.258
0 - train v loss: 0.0619 - val_test hybrid loss: 0.6361 - val_test m accuracy:
0.7993 - val_test v accuracy: 0.2570 - val_test m loss: 0.5626 - val_test v lo
ss: 0.0735
Epoch 15/100
76/76 [==============================] - 1s 7ms/step - train hybrid loss: 0.29
44 - train m accuracy: 0.9112 - train v accuracy: 0.2611 - train m loss: 0.234
9 - train v loss: 0.0595 - val_test hybrid loss: 0.6254 - val_test m accuracy:
0.7960 - val_test v accuracy: 0.2570 - val_test m loss: 0.5542 - val_test v lo
ss: 0.0712
Epoch 16/100
76/76 [==============================] - 1s 7ms/step - train hybrid loss: 0.25
97 - train m accuracy: 0.9232 - train v accuracy: 0.2470 - train m loss: 0.203
1 - train v loss: 0.0566 - val_test hybrid loss: 0.6191 - val_test m accuracy:
0.7993 - val_test v accuracy: 0.2570 - val_test m loss: 0.5542 - val_test v lo
ss: 0.0649
Epoch 17/100
76/76 [==============================] - 1s 7ms/step - train hybrid loss: 0.24
69 - train m accuracy: 0.9215 - train v accuracy: 0.2545 - train m loss: 0.195
7 - train v loss: 0.0513 - val_test hybrid loss: 0.5848 - val_test m accuracy:
0.8076 - val_test v accuracy: 0.2570 - val_test m loss: 0.5241 - val_test v lo
ss: 0.0607
Epoch 18/100
76/76 [==============================] - 1s 7ms/step - train hybrid loss: 0.22
17 - train m accuracy: 0.9381 - train v accuracy: 0.2556 - train m loss: 0.173
8 - train v loss: 0.0479 - val_test hybrid loss: 0.5918 - val_test m accuracy:
0.8027 - val_test v accuracy: 0.2570 - val_test m loss: 0.5305 - val_test v lo
ss: 0.0613
Epoch 19/100
76/76 [==============================] - 1s 7ms/step - train hybrid loss: 0.18
89 - train m accuracy: 0.9527 - train v accuracy: 0.2705 - train m loss: 0.143
8 - train v loss: 0.0451 - val_test hybrid loss: 0.6042 - val_test m accuracy:
0.8126 - val_test v accuracy: 0.2570 - val_test m loss: 0.5437 - val_test v lo
ss: 0.0605
Epoch 20/100
76/76 [==============================] - 1s 9ms/step - train hybrid loss: 0.18
89 - train m accuracy: 0.9452 - train v accuracy: 0.2694 - train m loss: 0.144
7 - train v loss: 0.0442 - val_test hybrid loss: 0.6147 - val_test m accuracy:
0.7960 - val_test v accuracy: 0.2570 - val_test m loss: 0.5510 - val_test v lo
ss: 0.0637
```

```
Epoch 21/100
76/76 [==============================] - 1s 7ms/step - train hybrid loss: 0.16
77 - train m accuracy: 0.9556 - train v accuracy: 0.2588 - train m loss: 0.127
1 - train v loss: 0.0406 - val_test hybrid loss: 0.6734 - val_test m accuracy:
0.7894 - val_test v accuracy: 0.2570 - val_test m loss: 0.6169 - val_test v lo
ss: 0.0565
Epoch 22/100
76/76 [==============================] - 1s 8ms/step - train hybrid loss: 0.14
86 - train m accuracy: 0.9631 - train v accuracy: 0.2708 - train m loss: 0.109
2 - train v loss: 0.0394 - val_test hybrid loss: 0.7004 - val_test m accuracy:
0.7944 - val_test v accuracy: 0.2570 - val_test m loss: 0.6436 - val_test v lo
ss: 0.0568
Epoch 23/100
76/76 [==============================] - 1s 9ms/step - train hybrid loss: 0.15
51 - train m accuracy: 0.9597 - train v accuracy: 0.2521 - train m loss: 0.117
5 - train v loss: 0.0376 - val_test hybrid loss: 0.6545 - val_test m accuracy:
0.8027 - val_test v accuracy: 0.2570 - val_test m loss: 0.5989 - val_test v lo
ss: 0.0557
Epoch 24/100
76/76 [==============================] - 1s 7ms/step - train hybrid loss: 0.13
89 - train m accuracy: 0.9651 - train v accuracy: 0.2587 - train m loss: 0.102
4 - train v loss: 0.0365 - val_test hybrid loss: 0.6994 - val_test m accuracy:
0.7910 - val_test v accuracy: 0.2570 - val_test m loss: 0.6444 - val_test v lo
ss: 0.0551
Epoch 25/100
76/76 [==============================] - 1s 7ms/step - train hybrid loss: 0.09
82 - train m accuracy: 0.9842 - train v accuracy: 0.2553 - train m loss: 0.065
7 - train v loss: 0.0325 - val_test hybrid loss: 0.6683 - val_test m accuracy:
0.8093 - val_test v accuracy: 0.2570 - val_test m loss: 0.6094 - val_test v lo
ss: 0.0588
Epoch 26/100
76/76 [==============================] - 1s 7ms/step - train hybrid loss: 0.08
40 - train m accuracy: 0.9892 - train v accuracy: 0.2425 - train m loss: 0.050
4 - train v loss: 0.0336 - val_test hybrid loss: 0.7238 - val_test m accuracy:
0.7944 - val_test v accuracy: 0.2570 - val_test m loss: 0.6637 - val_test v lo
ss: 0.0600
Epoch 27/100
76/76 [==============================] - 1s 7ms/step - train hybrid loss: 0.08
43 - train m accuracy: 0.9871 - train v accuracy: 0.2636 - train m loss: 0.052
4 - train v loss: 0.0319 - val_test hybrid loss: 0.6976 - val_test m accuracy:
0.8109 - val_test v accuracy: 0.2570 - val_test m loss: 0.6430 - val_test v lo
ss: 0.0546
Epoch 28/100
76/76 [==============================] - 1s 7ms/step - train hybrid loss: 0.07
36 - train m accuracy: 0.9896 - train v accuracy: 0.2620 - train m loss: 0.043
6 - train v loss: 0.0300 - val_test hybrid loss: 0.7505 - val_test m accuracy:
0.8027 - val_test v accuracy: 0.2570 - val_test m loss: 0.6940 - val_test v lo
ss: 0.0564
Epoch 29/100
76/76 [==============================] - 1s 7ms/step - train hybrid loss: 0.06
16 - train m accuracy: 0.9946 - train v accuracy: 0.2665 - train m loss: 0.033
0 - train v loss: 0.0285 - val_test hybrid loss: 0.7128 - val_test m accuracy:
0.8126 - val_test v accuracy: 0.2570 - val_test m loss: 0.6604 - val_test v lo
ss: 0.0524
Epoch 30/100
76/76 [==============================] - 1s 8ms/step - train hybrid loss: 0.06
86 - train m accuracy: 0.9909 - train v accuracy: 0.2577 - train m loss: 0.040
4 - train v loss: 0.0283 - val_test hybrid loss: 0.7513 - val_test m accuracy:
0.7993 - val_test v accuracy: 0.2570 - val_test m loss: 0.6916 - val_test v lo
ss: 0.0597
```

```
Epoch 31/100
76/76 [==============================] - 1s 7ms/step - train hybrid loss: 0.05
87 - train m accuracy: 0.9934 - train v accuracy: 0.2578 - train m loss: 0.032
4 - train v loss: 0.0263 - val_test hybrid loss: 0.8042 - val_test m accuracy:
0.8027 - val_test v accuracy: 0.2570 - val_test m loss: 0.7493 - val_test v lo
ss: 0.0549
Epoch 32/100
76/76 [==============================] - 1s 9ms/step - train hybrid loss: 0.05
59 - train m accuracy: 0.9954 - train v accuracy: 0.2776 - train m loss: 0.027
3 - train v loss: 0.0286 - val_test hybrid loss: 0.7510 - val_test m accuracy:
0.8126 - val_test v accuracy: 0.2570 - val_test m loss: 0.6969 - val_test v lo
ss: 0.0541
Epoch 33/100
76/76 [==============================] - 1s 7ms/step - train hybrid loss: 0.04
08 - train m accuracy: 0.9979 - train v accuracy: 0.2673 - train m loss: 0.017
1 - train v loss: 0.0238 - val_test hybrid loss: 0.7443 - val_test m accuracy:
0.8126 - val_test v accuracy: 0.2570 - val_test m loss: 0.6832 - val_test v lo
ss: 0.0612
Epoch 34/100
76/76 [==============================] - 1s 8ms/step - train hybrid loss: 0.03
83 - train m accuracy: 1.0000 - train v accuracy: 0.2503 - train m loss: 0.012
9 - train v loss: 0.0255 - val_test hybrid loss: 0.7327 - val_test m accuracy:
0.8342 - val_test v accuracy: 0.2570 - val_test m loss: 0.6741 - val_test v lo
ss: 0.0586
Epoch 35/100
76/76 [==============================] - 1s 7ms/step - train hybrid loss: 0.03
20 - train m accuracy: 1.0000 - train v accuracy: 0.2608 - train m loss: 0.011
4 - train v loss: 0.0206 - val_test hybrid loss: 0.7358 - val_test m accuracy:
0.8259 - val_test v accuracy: 0.2570 - val_test m loss: 0.6851 - val_test v lo
ss: 0.0507
Epoch 36/100
76/76 [==============================] - 1s 7ms/step - train hybrid loss: 0.02
68 - train m accuracy: 1.0000 - train v accuracy: 0.2693 - train m loss: 0.006
6 - train v loss: 0.0202 - val_test hybrid loss: 0.7426 - val_test m accuracy:
0.8109 - val_test v accuracy: 0.2570 - val_test m loss: 0.6913 - val_test v lo
ss: 0.0513
Epoch 37/100
76/76 [==============================] - 1s 8ms/step - train hybrid loss: 0.02
54 - train m accuracy: 1.0000 - train v accuracy: 0.2667 - train m loss: 0.005
7 - train v loss: 0.0197 - val_test hybrid loss: 0.7671 - val_test m accuracy:
0.8159 - val_test v accuracy: 0.2570 - val_test m loss: 0.7167 - val_test v lo
ss: 0.0504
Epoch 38/100
76/76 [==============================] - 1s 7ms/step - train hybrid loss: 0.02
25 - train m accuracy: 1.0000 - train v accuracy: 0.2546 - train m loss: 0.005
0 - train v loss: 0.0175 - val_test hybrid loss: 0.7774 - val_test m accuracy:
0.8159 - val_test v accuracy: 0.2570 - val_test m loss: 0.7297 - val_test v lo
ss: 0.0477
Epoch 39/100
76/76 [==============================] - 1s 7ms/step - train hybrid loss: 0.02
31 - train m accuracy: 1.0000 - train v accuracy: 0.2550 - train m loss: 0.004
8 - train v loss: 0.0183 - val_test hybrid loss: 0.8081 - val_test m accuracy:
0.8143 - val_test v accuracy: 0.2570 - val_test m loss: 0.7550 - val_test v lo
ss: 0.0531
Epoch 40/100
76/76 [==============================] - 1s 7ms/step - train hybrid loss: 0.02
02 - train m accuracy: 1.0000 - train v accuracy: 0.2592 - train m loss: 0.004
2 - train v loss: 0.0160 - val_test hybrid loss: 0.7838 - val_test m accuracy:
0.8093 - val_test v accuracy: 0.2570 - val_test m loss: 0.7358 - val_test v lo
ss: 0.0480
```

```
Epoch 41/100
76/76 [==============================] - 1s 8ms/step - train hybrid loss: 0.01
76 - train m accuracy: 1.0000 - train v accuracy: 0.2588 - train m loss: 0.003
9 - train v loss: 0.0137 - val_test hybrid loss: 0.7982 - val_test m accuracy:
0.8159 - val_test v accuracy: 0.2570 - val_test m loss: 0.7512 - val_test v lo
ss: 0.0471
Epoch 42/100
76/76 [==============================] - 1s 7ms/step - train hybrid loss: 0.01
81 - train m accuracy: 1.0000 - train v accuracy: 0.2543 - train m loss: 0.003
7 - train v loss: 0.0144 - val_test hybrid loss: 0.7890 - val_test m accuracy:
0.8176 - val_test v accuracy: 0.2570 - val_test m loss: 0.7421 - val_test v lo
ss: 0.0469
Epoch 43/100
76/76 [==============================] - 1s 7ms/step - train hybrid loss: 0.01
66 - train m accuracy: 1.0000 - train v accuracy: 0.2706 - train m loss: 0.003
4 - train v loss: 0.0131 - val_test hybrid loss: 0.8029 - val_test m accuracy:
0.8093 - val_test v accuracy: 0.2570 - val_test m loss: 0.7536 - val_test v lo
ss: 0.0494
Epoch 44/100
76/76 [==============================] - 1s 9ms/step - train hybrid loss: 0.01
63 - train m accuracy: 1.0000 - train v accuracy: 0.2509 - train m loss: 0.003
3 - train v loss: 0.0130 - val_test hybrid loss: 0.7974 - val_test m accuracy:
0.8192 - val_test v accuracy: 0.2570 - val_test m loss: 0.7471 - val_test v lo
ss: 0.0503
Epoch 45/100
76/76 [==============================] - 1s 7ms/step - train hybrid loss: 0.01
51 - train m accuracy: 1.0000 - train v accuracy: 0.2605 - train m loss: 0.003
0 - train v loss: 0.0121 - val_test hybrid loss: 0.8124 - val_test m accuracy:
0.8126 - val_test v accuracy: 0.2570 - val_test m loss: 0.7635 - val_test v lo
ss: 0.0489
Epoch 46/100
76/76 [==============================] - 1s 8ms/step - train hybrid loss: 0.01
35 - train m accuracy: 1.0000 - train v accuracy: 0.2503 - train m loss: 0.002
8 - train v loss: 0.0108 - val_test hybrid loss: 0.8227 - val_test m accuracy:
0.8109 - val_test v accuracy: 0.2570 - val_test m loss: 0.7710 - val_test v lo
ss: 0.0517
Epoch 47/100
76/76 [==============================] - 1s 7ms/step - train hybrid loss: 0.01
64 - train m accuracy: 1.0000 - train v accuracy: 0.2412 - train m loss: 0.002
9 - train v loss: 0.0135 - val_test hybrid loss: 0.8341 - val_test m accuracy:
0.8159 - val_test v accuracy: 0.2570 - val_test m loss: 0.7855 - val_test v lo
ss: 0.0486
Epoch 48/100
76/76 [==============================] - 1s 8ms/step - train hybrid loss: 0.01
46 - train m accuracy: 1.0000 - train v accuracy: 0.2689 - train m loss: 0.002
7 - train v loss: 0.0120 - val_test hybrid loss: 0.8266 - val_test m accuracy:
0.8176 - val_test v accuracy: 0.2570 - val_test m loss: 0.7799 - val_test v lo
ss: 0.0467
Epoch 49/100
76/76 [==============================] - 1s 8ms/step - train hybrid loss: 0.01
49 - train m accuracy: 1.0000 - train v accuracy: 0.2555 - train m loss: 0.002
6 - train v loss: 0.0124 - val_test hybrid loss: 0.8444 - val_test m accuracy:
0.8109 - val_test v accuracy: 0.2570 - val_test m loss: 0.7950 - val_test v lo
ss: 0.0494
Epoch 50/100
76/76 [==============================] - 1s 7ms/step - train hybrid loss: 0.01
61 - train m accuracy: 1.0000 - train v accuracy: 0.2606 - train m loss: 0.002
7 - train v loss: 0.0135 - val_test hybrid loss: 0.8569 - val_test m accuracy:
0.8109 - val_test v accuracy: 0.2570 - val_test m loss: 0.8080 - val_test v lo
ss: 0.0490
```

```
Epoch 51/100
76/76 [==============================] - 1s 7ms/step - train hybrid loss: 0.01
37 - train m accuracy: 1.0000 - train v accuracy: 0.2707 - train m loss: 0.002
4 - train v loss: 0.0113 - val_test hybrid loss: 0.8436 - val_test m accuracy:
0.8159 - val_test v accuracy: 0.2570 - val_test m loss: 0.7986 - val_test v lo
ss: 0.0449
Epoch 52/100
76/76 [==============================] - 1s 7ms/step - train hybrid loss: 0.01
32 - train m accuracy: 1.0000 - train v accuracy: 0.2576 - train m loss: 0.002
2 - train v loss: 0.0110 - val_test hybrid loss: 0.8605 - val_test m accuracy:
0.8076 - val_test v accuracy: 0.2570 - val_test m loss: 0.8113 - val_test v lo
ss: 0.0492
Epoch 53/100
76/76 [==============================] - 1s 7ms/step - train hybrid loss: 0.01
25 - train m accuracy: 1.0000 - train v accuracy: 0.2701 - train m loss: 0.002
2 - train v loss: 0.0103 - val_test hybrid loss: 0.8611 - val_test m accuracy:
0.8126 - val_test v accuracy: 0.2570 - val_test m loss: 0.8124 - val_test v lo
ss: 0.0488
Epoch 54/100
76/76 [==============================] - 1s 7ms/step - train hybrid loss: 0.01
31 - train m accuracy: 1.0000 - train v accuracy: 0.2671 - train m loss: 0.002
2 - train v loss: 0.0109 - val_test hybrid loss: 0.8603 - val_test m accuracy:
0.8143 - val_test v accuracy: 0.2570 - val_test m loss: 0.8135 - val_test v lo
ss: 0.0468
Epoch 55/100
76/76 [==============================] - 1s 8ms/step - train hybrid loss: 0.01
52 - train m accuracy: 1.0000 - train v accuracy: 0.2378 - train m loss: 0.002
2 - train v loss: 0.0130 - val_test hybrid loss: 0.9058 - val_test m accuracy:
0.8076 - val_test v accuracy: 0.2570 - val_test m loss: 0.8534 - val_test v lo
ss: 0.0524
Epoch 56/100
76/76 [==============================] - 1s 8ms/step - train hybrid loss: 0.01
19 - train m accuracy: 1.0000 - train v accuracy: 0.2490 - train m loss: 0.002
2 - train v loss: 0.0097 - val_test hybrid loss: 0.8978 - val_test m accuracy:
0.8109 - val_test v accuracy: 0.2570 - val_test m loss: 0.8537 - val_test v lo
ss: 0.0441
Epoch 57/100
76/76 [==============================] - 1s 7ms/step - train hybrid loss: 0.01
02 - train m accuracy: 1.0000 - train v accuracy: 0.2681 - train m loss: 0.001
8 - train v loss: 0.0084 - val_test hybrid loss: 0.8990 - val_test m accuracy:
0.8176 - val_test v accuracy: 0.2570 - val_test m loss: 0.8513 - val_test v lo
ss: 0.0476
Epoch 58/100
76/76 [==============================] - 1s 7ms/step - train hybrid loss: 0.01
03 - train m accuracy: 1.0000 - train v accuracy: 0.2711 - train m loss: 0.001
7 - train v loss: 0.0086 - val_test hybrid loss: 0.9011 - val_test m accuracy:
0.8143 - val_test v accuracy: 0.2570 - val_test m loss: 0.8531 - val_test v lo
ss: 0.0480
Epoch 59/100
76/76 [==============================] - 1s 7ms/step - train hybrid loss: 0.01
37 - train m accuracy: 1.0000 - train v accuracy: 0.2560 - train m loss: 0.001
9 - train v loss: 0.0118 - val_test hybrid loss: 0.8827 - val_test m accuracy:
0.8093 - val_test v accuracy: 0.2570 - val_test m loss: 0.8337 - val_test v lo
ss: 0.0489
Epoch 60/100
76/76 [==============================] - 1s 7ms/step - train hybrid loss: 0.01
28 - train m accuracy: 1.0000 - train v accuracy: 0.2583 - train m loss: 0.001
8 - train v loss: 0.0110 - val_test hybrid loss: 0.9038 - val_test m accuracy:
0.8159 - val_test v accuracy: 0.2570 - val_test m loss: 0.8568 - val_test v lo
ss: 0.0470
```

```
Epoch 61/100
76/76 [==============================] - 1s 7ms/step - train hybrid loss: 0.07
30 - train m accuracy: 0.9875 - train v accuracy: 0.2420 - train m loss: 0.056
4 - train v loss: 0.0166 - val_test hybrid loss: 1.9856 - val_test m accuracy:
0.7015 - val_test v accuracy: 0.2570 - val_test m loss: 1.9249 - val_test v lo
ss: 0.0608
Epoch 62/100
76/76 [==============================] - 1s 8ms/step - train hybrid loss: 0.54
14 - train m accuracy: 0.8381 - train v accuracy: 0.2596 - train m loss: 0.489
7 - train v loss: 0.0517 - val_test hybrid loss: 0.7678 - val_test m accuracy:
0.7678 - val_test v accuracy: 0.2570 - val_test m loss: 0.7085 - val_test v lo
ss: 0.0593
Epoch 63/100
76/76 [==============================] - 1s 8ms/step - train hybrid loss: 0.23
23 - train m accuracy: 0.9240 - train v accuracy: 0.2680 - train m loss: 0.198
4 - train v loss: 0.0338 - val_test hybrid loss: 0.9541 - val_test m accuracy:
0.7579 - val_test v accuracy: 0.2570 - val_test m loss: 0.8938 - val_test v lo
ss: 0.0603
Epoch 64/100
76/76 [==============================] - 1s 7ms/step - train hybrid loss: 0.20
44 - train m accuracy: 0.9411 - train v accuracy: 0.2406 - train m loss: 0.177
6 - train v loss: 0.0268 - val_test hybrid loss: 0.7839 - val_test m accuracy:
0.8010 - val_test v accuracy: 0.2570 - val_test m loss: 0.7290 - val_test v lo
ss: 0.0549
Epoch 65/100
76/76 [==============================] - 1s 8ms/step - train hybrid loss: 0.12
75 - train m accuracy: 0.9685 - train v accuracy: 0.2456 - train m loss: 0.102
0 - train v loss: 0.0255 - val_test hybrid loss: 0.8001 - val_test m accuracy:
0.8109 - val_test v accuracy: 0.2570 - val_test m loss: 0.7478 - val_test v lo
ss: 0.0522
Epoch 66/100
76/76 [==============================] - 1s 7ms/step - train hybrid loss: 0.06
46 - train m accuracy: 0.9863 - train v accuracy: 0.2574 - train m loss: 0.043
3 - train v loss: 0.0213 - val_test hybrid loss: 0.7892 - val_test m accuracy:
0.8176 - val_test v accuracy: 0.2570 - val_test m loss: 0.7373 - val_test v lo
ss: 0.0519
Epoch 67/100
76/76 [==============================] - 1s 8ms/step - train hybrid loss: 0.03
77 - train m accuracy: 0.9958 - train v accuracy: 0.2481 - train m loss: 0.021
8 - train v loss: 0.0158 - val_test hybrid loss: 0.8198 - val_test m accuracy:
0.8159 - val_test v accuracy: 0.2570 - val_test m loss: 0.7724 - val_test v lo
ss: 0.0474
Epoch 68/100
76/76 [==============================] - 1s 9ms/step - train hybrid loss: 0.02
15 - train m accuracy: 1.0000 - train v accuracy: 0.2843 - train m loss: 0.008
2 - train v loss: 0.0133 - val_test hybrid loss: 0.8722 - val_test m accuracy:
0.8143 - val_test v accuracy: 0.2570 - val_test m loss: 0.8218 - val_test v lo
ss: 0.0504
Epoch 69/100
76/76 [==============================] - 1s 7ms/step - train hybrid loss: 0.01
66 - train m accuracy: 1.0000 - train v accuracy: 0.2713 - train m loss: 0.004
4 - train v loss: 0.0122 - val_test hybrid loss: 0.8576 - val_test m accuracy:
0.8176 - val_test v accuracy: 0.2570 - val_test m loss: 0.8097 - val_test v lo
ss: 0.0479
Epoch 70/100
76/76 [==============================] - 1s 9ms/step - train hybrid loss: 0.01
19 - train m accuracy: 1.0000 - train v accuracy: 0.2429 - train m loss: 0.002
7 - train v loss: 0.0091 - val_test hybrid loss: 0.8580 - val_test m accuracy:
0.8192 - val_test v accuracy: 0.2570 - val_test m loss: 0.8115 - val_test v lo
ss: 0.0464
```

```
Epoch 71/100
76/76 [==============================] - 1s 8ms/step - train hybrid loss: 0.01
02 - train m accuracy: 1.0000 - train v accuracy: 0.2648 - train m loss: 0.002
3 - train v loss: 0.0079 - val_test hybrid loss: 0.8764 - val_test m accuracy:
0.8126 - val_test v accuracy: 0.2570 - val_test m loss: 0.8309 - val_test v lo
ss: 0.0455
Epoch 72/100
76/76 [==============================] - 1s 8ms/step - train hybrid loss: 0.00
85 - train m accuracy: 1.0000 - train v accuracy: 0.2425 - train m loss: 0.002
0 - train v loss: 0.0065 - val_test hybrid loss: 0.8889 - val_test m accuracy:
0.8143 - val_test v accuracy: 0.2570 - val_test m loss: 0.8426 - val_test v lo
ss: 0.0462
Epoch 73/100
76/76 [==============================] - 1s 8ms/step - train hybrid loss: 0.00
75 - train m accuracy: 1.0000 - train v accuracy: 0.2581 - train m loss: 0.001
7 - train v loss: 0.0058 - val_test hybrid loss: 0.8917 - val_test m accuracy:
0.8126 - val_test v accuracy: 0.2570 - val_test m loss: 0.8424 - val_test v lo
ss: 0.0493
Epoch 74/100
76/76 [==============================] - 1s 9ms/step - train hybrid loss: 0.00
73 - train m accuracy: 1.0000 - train v accuracy: 0.2482 - train m loss: 0.001
6 - train v loss: 0.0057 - val_test hybrid loss: 0.8999 - val_test m accuracy:
0.8192 - val_test v accuracy: 0.2570 - val_test m loss: 0.8535 - val_test v lo
ss: 0.0464
Epoch 75/100
76/76 [==============================] - 1s 8ms/step - train hybrid loss: 0.00
65 - train m accuracy: 1.0000 - train v accuracy: 0.2604 - train m loss: 0.001
5 - train v loss: 0.0050 - val_test hybrid loss: 0.9019 - val_test m accuracy:
0.8109 - val_test v accuracy: 0.2570 - val_test m loss: 0.8545 - val_test v lo
ss: 0.0474
Epoch 76/100
76/76 [==============================] - 1s 7ms/step - train hybrid loss: 0.00
69 - train m accuracy: 1.0000 - train v accuracy: 0.2585 - train m loss: 0.001
4 - train v loss: 0.0055 - val_test hybrid loss: 0.9114 - val_test m accuracy:
0.8126 - val_test v accuracy: 0.2570 - val_test m loss: 0.8656 - val_test v lo
ss: 0.0458
Epoch 77/100
76/76 [==============================] - 1s 8ms/step - train hybrid loss: 0.00
59 - train m accuracy: 1.0000 - train v accuracy: 0.2635 - train m loss: 0.001
3 - train v loss: 0.0046 - val_test hybrid loss: 0.9236 - val_test m accuracy:
0.8143 - val_test v accuracy: 0.2570 - val_test m loss: 0.8770 - val_test v lo
ss: 0.0467
Epoch 78/100
76/76 [==============================] - 1s 8ms/step - train hybrid loss: 0.00
56 - train m accuracy: 1.0000 - train v accuracy: 0.2600 - train m loss: 0.001
2 - train v loss: 0.0044 - val_test hybrid loss: 0.9384 - val_test m accuracy:
0.8109 - val_test v accuracy: 0.2570 - val_test m loss: 0.8912 - val_test v lo
ss: 0.0473
Epoch 79/100
76/76 [==============================] - 1s 8ms/step - train hybrid loss: 0.00
53 - train m accuracy: 1.0000 - train v accuracy: 0.2526 - train m loss: 0.001
1 - train v loss: 0.0042 - val_test hybrid loss: 0.9380 - val_test m accuracy:
0.8192 - val_test v accuracy: 0.2570 - val_test m loss: 0.8934 - val_test v lo
ss: 0.0445
Epoch 80/100
76/76 [==============================] - 1s 9ms/step - train hybrid loss: 0.00
47 - train m accuracy: 1.0000 - train v accuracy: 0.2539 - train m loss: 0.001
1 - train v loss: 0.0037 - val_test hybrid loss: 0.9464 - val_test m accuracy:
0.8159 - val_test v accuracy: 0.2570 - val_test m loss: 0.8991 - val_test v lo
ss: 0.0474
```

```
Epoch 81/100
76/76 [==============================] - 1s 8ms/step - train hybrid loss: 0.00
43 - train m accuracy: 1.0000 - train v accuracy: 0.2699 - train m loss: 9.271
9e-04 - train v loss: 0.0033 - val_test hybrid loss: 0.9419 - val_test m accur
acy: 0.8192 - val_test v accuracy: 0.2570 - val_test m loss: 0.8973 - val_test
v loss: 0.0446
Epoch 82/100
76/76 [==============================] - 1s 8ms/step - train hybrid loss: 0.00
43 - train m accuracy: 1.0000 - train v accuracy: 0.2555 - train m loss: 9.228
3e-04 - train v loss: 0.0034 - val_test hybrid loss: 0.9622 - val_test m accur
acy: 0.8176 - val_test v accuracy: 0.2570 - val_test m loss: 0.9160 - val_test
v loss: 0.0462
Epoch 83/100
76/76 [==============================] - 1s 8ms/step - train hybrid loss: 0.00
41 - train m accuracy: 1.0000 - train v accuracy: 0.2531 - train m loss: 8.438
0e-04 - train v loss: 0.0032 - val_test hybrid loss: 0.9695 - val_test m accur
acy: 0.8192 - val_test v accuracy: 0.2570 - val_test m loss: 0.9237 - val_test
v loss: 0.0457
Epoch 84/100
76/76 [==============================] - 1s 8ms/step - train hybrid loss: 0.00
51 - train m accuracy: 1.0000 - train v accuracy: 0.2606 - train m loss: 8.507
4e-04 - train v loss: 0.0043 - val_test hybrid loss: 0.9563 - val_test m accur
acy: 0.8109 - val_test v accuracy: 0.2570 - val_test m loss: 0.9085 - val_test
v loss: 0.0479
Epoch 85/100
76/76 [==============================] - 1s 8ms/step - train hybrid loss: 0.00
62 - train m accuracy: 1.0000 - train v accuracy: 0.2456 - train m loss: 8.238
3e-04 - train v loss: 0.0054 - val_test hybrid loss: 0.9679 - val_test m accur
acy: 0.8043 - val_test v accuracy: 0.2570 - val_test m loss: 0.9211 - val_test
v loss: 0.0467
Epoch 86/100
76/76 [==============================] - 1s 8ms/step - train hybrid loss: 0.00
68 - train m accuracy: 1.0000 - train v accuracy: 0.2558 - train m loss: 8.614
3e-04 - train v loss: 0.0060 - val_test hybrid loss: 0.9612 - val_test m accur
acy: 0.8060 - val_test v accuracy: 0.2570 - val_test m loss: 0.9161 - val_test
v loss: 0.0451
Epoch 87/100
76/76 [==============================] - 1s 8ms/step - train hybrid loss: 0.00
64 - train m accuracy: 1.0000 - train v accuracy: 0.2640 - train m loss: 8.875
1e-04 - train v loss: 0.0055 - val_test hybrid loss: 0.9596 - val_test m accur
acy: 0.8093 - val_test v accuracy: 0.2570 - val_test m loss: 0.9129 - val_test
v loss: 0.0467
Epoch 88/100
76/76 [==============================] - 1s 8ms/step - train hybrid loss: 0.00
52 - train m accuracy: 1.0000 - train v accuracy: 0.2601 - train m loss: 7.751
1e-04 - train v loss: 0.0044 - val_test hybrid loss: 0.9546 - val_test m accur
acy: 0.8126 - val_test v accuracy: 0.2570 - val_test m loss: 0.9074 - val_test
v loss: 0.0472
Epoch 89/100
76/76 [==============================] - 1s 8ms/step - train hybrid loss: 0.00
52 - train m accuracy: 1.0000 - train v accuracy: 0.2615 - train m loss: 7.660
3e-04 - train v loss: 0.0044 - val_test hybrid loss: 0.9779 - val_test m accur
acy: 0.8159 - val_test v accuracy: 0.2570 - val_test m loss: 0.9321 - val_test
v loss: 0.0458
Epoch 90/100
76/76 [==============================] - 1s 8ms/step - train hybrid loss: 0.00
70 - train m accuracy: 1.0000 - train v accuracy: 0.2744 - train m loss: 7.398
2e-04 - train v loss: 0.0062 - val_test hybrid loss: 0.9862 - val_test m accur
acy: 0.8109 - val_test v accuracy: 0.2570 - val_test m loss: 0.9366 - val_test
v loss: 0.0496
```

```
Epoch 91/100
76/76 [==============================] - 1s 8ms/step - train hybrid loss: 0.00
69 - train m accuracy: 1.0000 - train v accuracy: 0.2513 - train m loss: 7.908
8e-04 - train v loss: 0.0061 - val_test hybrid loss: 0.9660 - val_test m accur
acy: 0.8126 - val_test v accuracy: 0.2570 - val_test m loss: 0.9194 - val_test
v loss: 0.0466
Epoch 92/100
76/76 [==============================] - 1s 8ms/step - train hybrid loss: 0.00
52 - train m accuracy: 1.0000 - train v accuracy: 0.2711 - train m loss: 7.459
2e-04 - train v loss: 0.0045 - val_test hybrid loss: 0.9809 - val_test m accur
acy: 0.8109 - val_test v accuracy: 0.2570 - val_test m loss: 0.9377 - val_test
v loss: 0.0432
Epoch 93/100
76/76 [==============================] - 1s 10ms/step - train hybrid loss: 0.0
053 - train m accuracy: 1.0000 - train v accuracy: 0.2525 - train m loss: 6.77
66e-04 - train v loss: 0.0046 - val_test hybrid loss: 0.9879 - val_test m accu
racy: 0.8093 - val_test v accuracy: 0.2570 - val_test m loss: 0.9429 - val_tes
t v loss: 0.0450
Epoch 94/100
76/76 [==============================] - 1s 9ms/step - train hybrid loss: 0.00
60 - train m accuracy: 1.0000 - train v accuracy: 0.2649 - train m loss: 6.751
9e-04 - train v loss: 0.0053 - val_test hybrid loss: 0.9971 - val_test m accur
acy: 0.8159 - val_test v accuracy: 0.2570 - val_test m loss: 0.9509 - val_test
v loss: 0.0462
Epoch 95/100
76/76 [==============================] - 1s 8ms/step - train hybrid loss: 0.00
51 - train m accuracy: 1.0000 - train v accuracy: 0.2521 - train m loss: 6.233
5e-04 - train v loss: 0.0045 - val_test hybrid loss: 1.0070 - val_test m accur
acy: 0.8126 - val_test v accuracy: 0.2570 - val_test m loss: 0.9622 - val_test
v loss: 0.0448
Epoch 96/100
76/76 [==============================] - 1s 8ms/step - train hybrid loss: 0.00
77 - train m accuracy: 1.0000 - train v accuracy: 0.2488 - train m loss: 7.230
5e-04 - train v loss: 0.0070 - val_test hybrid loss: 0.9650 - val_test m accur
acy: 0.8192 - val_test v accuracy: 0.2570 - val_test m loss: 0.9185 - val_test
v loss: 0.0465
Epoch 97/100
76/76 [==============================] - 1s 13ms/step - train hybrid loss: 0.0
059 - train m accuracy: 1.0000 - train v accuracy: 0.2681 - train m loss: 7.18
94e-04 - train v loss: 0.0052 - val_test hybrid loss: 0.9811 - val_test m accu
racy: 0.8176 - val_test v accuracy: 0.2570 - val_test m loss: 0.9322 - val_tes
t v loss: 0.0489
Epoch 98/100
76/76 [==============================] - 1s 10ms/step - train hybrid loss: 0.0
085 - train m accuracy: 1.0000 - train v accuracy: 0.2540 - train m loss: 8.00
41e-04 - train v loss: 0.0077 - val_test hybrid loss: 0.9580 - val_test m accu
racy: 0.8176 - val_test v accuracy: 0.2570 - val_test m loss: 0.9069 - val_tes
t v loss: 0.0511
Epoch 99/100
76/76 [==============================] - 1s 8ms/step - train hybrid loss: 0.00
76 - train m accuracy: 1.0000 - train v accuracy: 0.2571 - train m loss: 8.584
2e-04 - train v loss: 0.0067 - val_test hybrid loss: 0.9478 - val_test m accur
acy: 0.8176 - val_test v accuracy: 0.2570 - val_test m loss: 0.9013 - val_test
v loss: 0.0465
Epoch 100/100
76/76 [==============================] - 1s 8ms/step - train hybrid loss: 0.00
60 - train m accuracy: 1.0000 - train v accuracy: 0.2588 - train m loss: 7.014
6e-04 - train v loss: 0.0053 - val_test hybrid loss: 0.9968 - val_test m accur
acy: 0.8176 - val_test v accuracy: 0.2570 - val_test m loss: 0.9512 - val_test
v loss: 0.0457
```

Visualization

Get the loss and accuracy on training set and validation set by accessing model.history.history

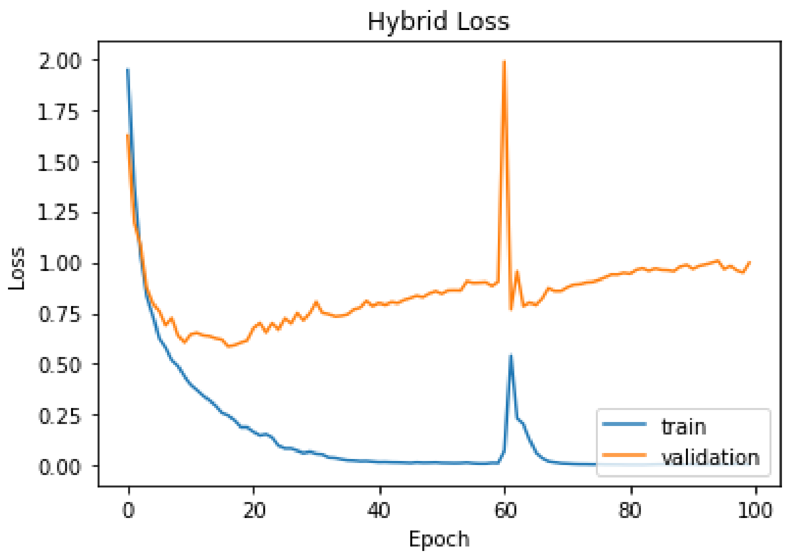
Plot the two loss curves where the x axis is the number of epochs and y axis is the loss.

Plot the two accuracy curves where the x-axis is the number of epochs and y-axis is the accuracy.

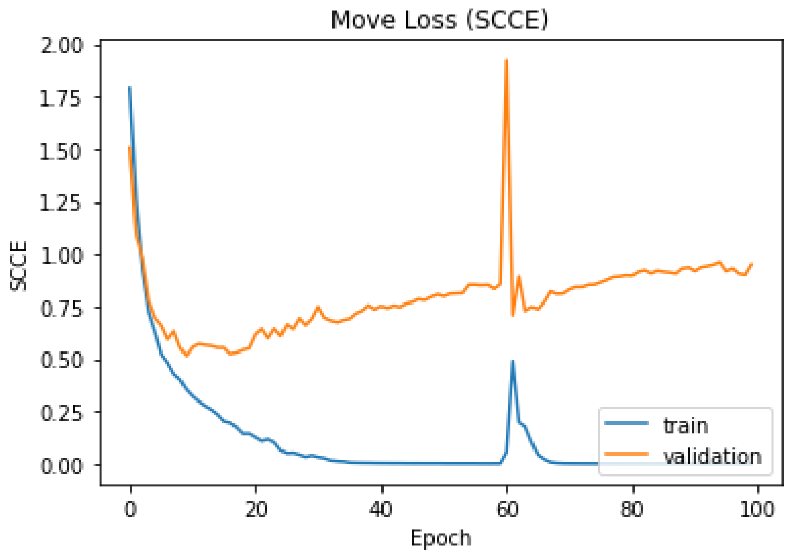
In [9]:
```
list(history.history.keys())
```

Out[9]:
```
['train hybrid loss',
 'train m accuracy',
 'train v accuracy',
 'train m loss',
 'train v loss',
 'val_test hybrid loss',
 'val_test m accuracy',
 'val_test v accuracy',
 'val_test m loss',
 'val_test v loss']
```

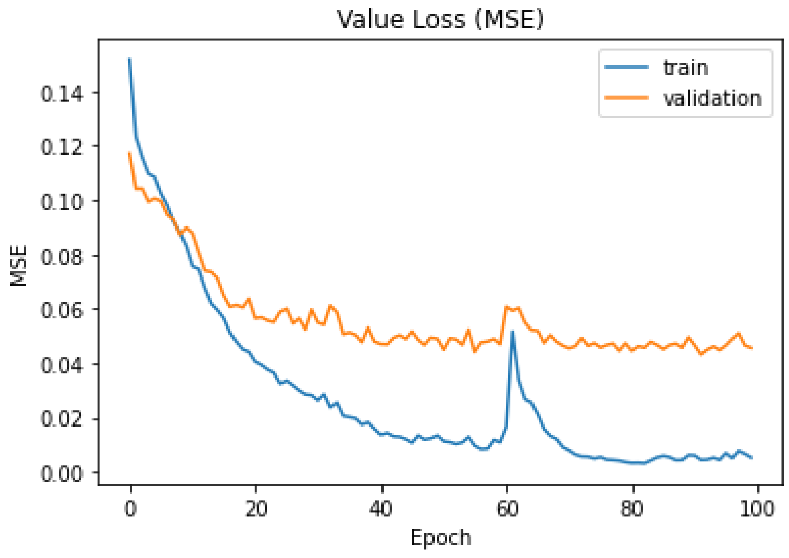
In [11]:
```
plt.plot(history.history['train hybrid loss'], label = 'train')
plt.plot(history.history['val_test hybrid loss'], label = 'validation')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend(loc='lower right')
plt.title('Hybrid Loss')
plt.show()
```



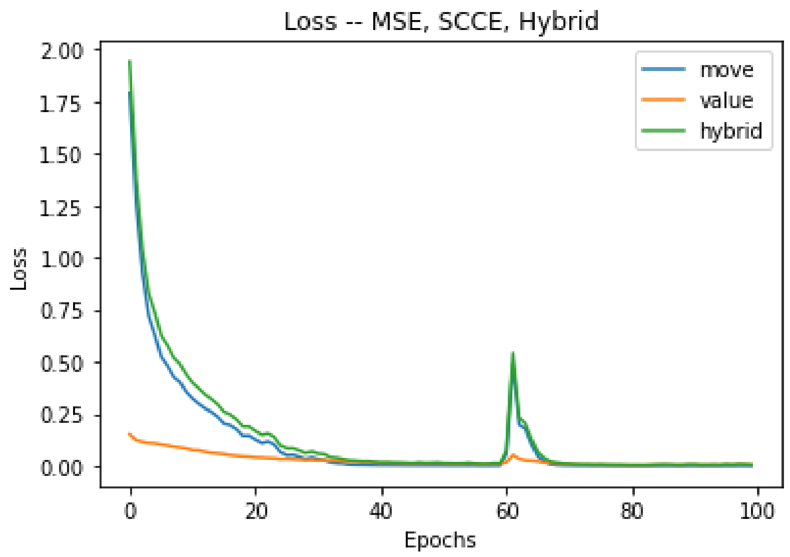
In [30]:
```
plt.plot(history.history['train m loss'],     label = 'train')
plt.plot(history.history['val_test m loss'], label = 'validation')
plt.xlabel('Epoch')
plt.ylabel('SCCE')
plt.legend(loc='lower right')
plt.title('Move Loss (SCCE)')
plt.show()
```



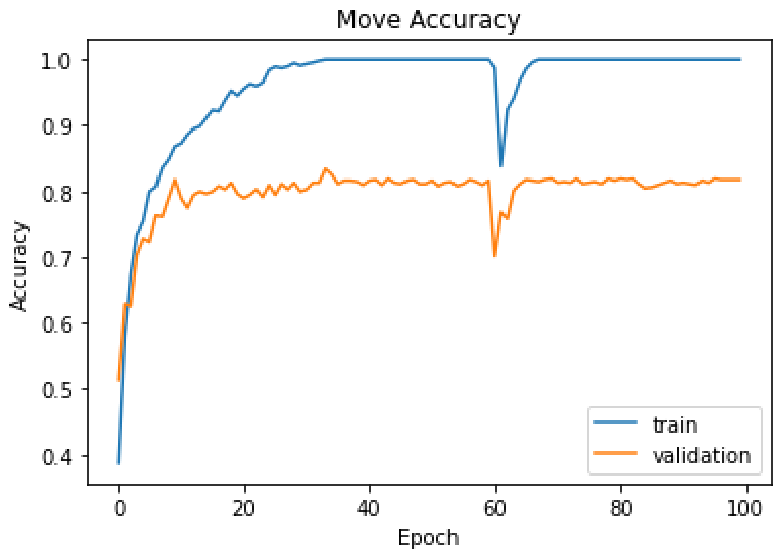
In [27]:
```
plt.plot(history.history['train v loss'],     label = 'train')
plt.plot(history.history['val_test v loss'], label = 'validation')
plt.xlabel('Epoch')
plt.ylabel('MSE')
plt.legend(loc='upper right')
plt.title('Value Loss (MSE)')
plt.show()
```



In [24]:
```python
plt.plot(history.history['train m loss'],      label = 'move')
plt.plot(history.history['train v loss'],      label = 'value')
plt.plot(history.history['train hybrid loss'], label = 'hybrid')
plt.title('Loss -- MSE, SCCE, Hybrid')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.show()
```



In [31]:
```python
plt.plot(history.history['train m accuracy'],      label = 'train')
plt.plot(history.history['val_test m accuracy'], label = 'validation')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.legend(loc='lower right')
plt.title('Move Accuracy')
plt.show()
```



Evaluation

Evaluate on the test set and print the test accuracy.

```
In [57]: m_pred, v_pred = model.predict(X_test)
```

```
In [34]: accuracy_score(y_move_test.ravel(), np.argmax(m_pred, axis=1))
```

```
Out[34]: 0.8188453881884539
```

```
In [35]: mean_squared_error(y_value_test, v_pred)
```

```
Out[35]: 0.04914453
```

Prediction

Pick a few game states from test set, and predict their next moves and outcomes. Do you think your model can make reasonable prediction for a given game state? Discuss your findings.

```
In [36]: def print_gn(x):
             cur_state = []
             for _ in range(3):
                 cur_state.append(3*['-'])
             O_pos_tuple = np.nonzero(x[0])
             for i, j in zip(O_pos_tuple[0], O_pos_tuple[1]):
                 cur_state[i][j] = 'O'

             X_pos_tuple = np.nonzero(x[1])
             for i, j in zip(X_pos_tuple[0], X_pos_tuple[1]):
                 cur_state[i][j] = 'X'
             cur_gn = MNKNode(cur_state)

             print("%s" % cur_gn)
             print("Next player: %s" % cur_gn.next_player())
```

```
In [37]: def print_gn_pred(x):
             x_new = np.expand_dims(x, axis=0)
             m_pred, v_pred = model.predict(x_new)
             m_pred_reshape = np.reshape(m_pred, (3, 3))

             print("Move probabilities:")
             for i in m_pred_reshape:
                 print(" ".join(['{:.2f}'.format(j) for j in i]))
             print()
             print("Value: %.4f" % v_pred[0][0])

         def print_gn_y(y_m,y_v):
             print("Move: %d" % y_m[0])
             print()
             print("Value: {}".format(y_v[0]))
```

RESULTS/DISCUSSION

In [38]:
```
import random

def random_results(X_test=X_test,number=10,seed=1234):
  random.seed(seed)
  j=0
  for i in random.sample(range(0, len(X_test)), number):
    hash_row = '#'*(len(str(j+1))+9)
    print('{}\n# TEST {} #\n{}'.format(hash_row,j+1,hash_row))
    print_gn(X_test[i])
    print_gn_pred(X_test[i])
    j+=1
    print('\n')

random_results()
```

```
##########
# TEST 1 #
##########
- X O
- O X
X O -

Next player: X
Move probabilities:
1.00 0.00 0.00
0.00 0.00 0.00
0.00 0.00 0.00

Value: 0.6399


##########
# TEST 2 #
##########
- X X
- X -
- O O

Next player: O
Move probabilities:
0.74 0.00 0.00
0.00 0.00 0.00
0.26 0.00 0.00

Value: 0.0327


##########
# TEST 3 #
##########
- - -
X - -
X O O

Next player: X
Move probabilities:
1.00 0.00 0.00
0.00 0.00 0.00
0.00 0.00 0.00

Value: 1.0000


##########
# TEST 4 #
##########
X O -
- O O
- X X

Next player: X
Move probabilities:
0.00 0.00 0.00
0.02 0.00 0.00
0.97 0.00 0.00
```

```
Value: 0.9964


##########
# TEST 5 #
##########
- O -
X X O
- X -

Next player: O
Move probabilities:
0.51 0.00 0.49
0.00 0.00 0.00
0.00 0.00 0.00

Value: 0.4336


##########
# TEST 6 #
##########
- X O
- - X
O O X

Next player: X
Move probabilities:
0.95 0.00 0.00
0.00 0.05 0.00
0.00 0.00 0.00

Value: 0.6505


##########
# TEST 7 #
##########
- X X
O - O
X O -

Next player: X
Move probabilities:
1.00 0.00 0.00
0.00 0.00 0.00
0.00 0.00 0.00

Value: 1.0000


##########
# TEST 8 #
##########
X - -
- O X
- O X

Next player: O
```

```
Move probabilities:
0.00 1.00 0.00
0.00 0.00 0.00
0.00 0.00 0.00

Value: 0.0210


##########
# TEST 9 #
##########
X X -
- - O
O O X

Next player: X
Move probabilities:
0.00 0.00 1.00
0.00 0.00 0.00
0.00 0.00 0.00

Value: 1.0000


###########
# TEST 10 #
###########
- - -
O O X
- - X

Next player: X
Move probabilities:
0.00 0.00 1.00
0.00 0.00 0.00
0.00 0.00 0.00

Value: 0.9999
```

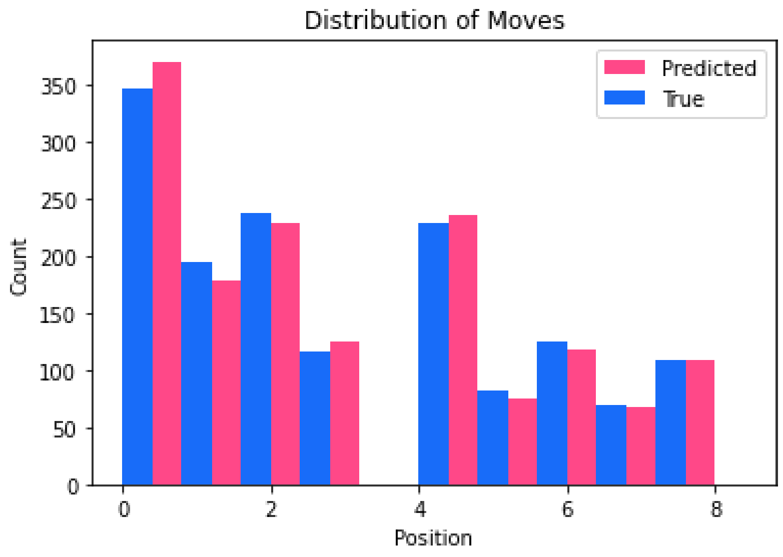
I intuitively reject 2/10 of the solutions above.

> Test 2 predicts a sub-optimal move -- O blocks a possible winning move for X, but fails to play a move that would result in an immediate win. The predicted value of this move is 0.0327, which makes sense, since it missed an opportunity to win the game.
>
> Test 8 results in a winning move, but the predicted value is only 0.0210, which doesn't make sense.

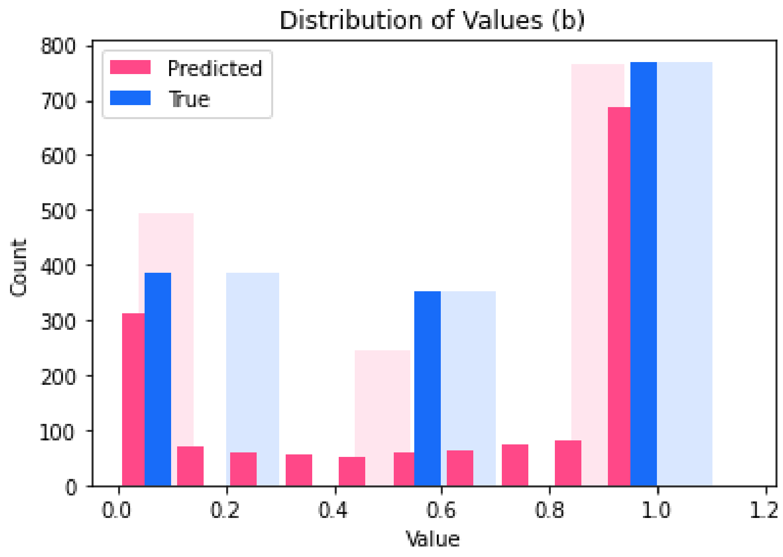
The other eight predictions seem to result in rational moves and predicted values that are within the appropriate range.

Given that there are two outputs -- move and value -- there are two ways each prediction could be accurate or inaccurate. I decided to plot the distributions of moves and values, including both the actual and predicted results.

In [183]:
```
import scipy
import matplotlib.mlab as mlab
from scipy.stats import norm

m_pred_max = [p.tolist().index(max(p)) for p in m_pred]

plt.hist(m_pred_max,       color='#FF4888', width=0.4, align='right',   label=
'Predicted')
plt.hist(y_move_test[:,0], color='#186CF9', width=0.4, align='mid', label='Tru
e')
plt.xlabel('Position')
plt.ylabel('Count')
plt.title('Distribution of Moves')
plt.legend()
plt.show()


plt.hist((v_pred[:,0],
          y_value_test[:,0]),
         width=0.1,
         align='mid',
         color=('#FFABC8','#84B2FF'),
         alpha=0.3,
         bins=[0,0.4, 0.8, 1.2])

plt.xlabel('Value')
plt.ylabel('Count')
plt.title('Distribution of Values (a)')
plt.legend()
#plt.show()

plt.hist((v_pred[:,0],
          y_value_test[:,0]),
         width=0.05,
         align='mid',
         color=('#FF4888','#186CF9'),
         label=('Predicted','True'))
         #bins=[0,0.2, 0.4,0.6, 0.8,1,1.2])

plt.xlabel('Value')
plt.ylabel('Count')
plt.title('Distribution of Values (b)')
plt.legend()
plt.show()
```



```
No handles with labels found to put in legend.
```

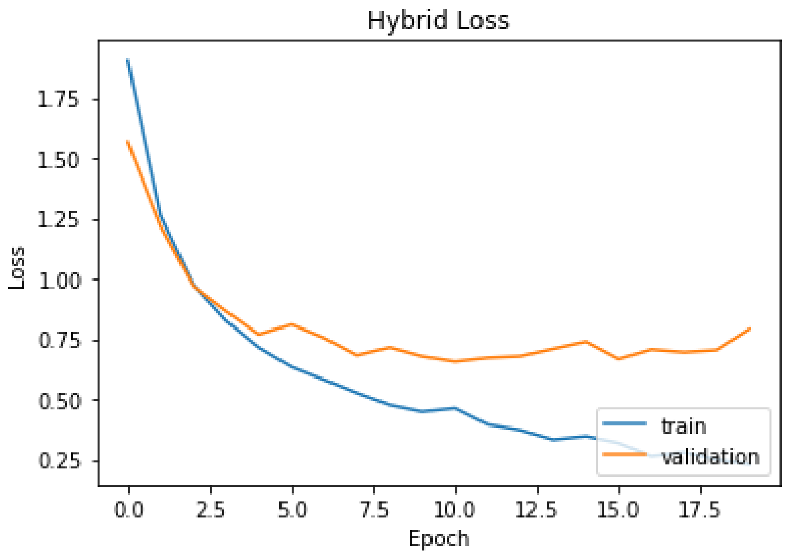


The actual and predicted results have closely matching distributions of moves. The distributions of values are not as close, as the predicted results skew toward the 0 and 1, which a large portion of the actual values are 0.5. This may be a better representation of the utility of possible moves, since it allows for intermediate values, but this is not reflected in the 'true' data.

One last note worth making is that the validation loss increases much earlier than 100 epochs. It is possible that model is overfit. We can see how the results would change by limiting the model to training on 20 epochs.

In [185]:
```
model = ConvResNN()
optimizer = Adam(1e-3)
model.compile(optimizer)

start_time = time()
history2 = model.fit(X_train, [y_move_train, y_value_train],
                    epochs=20,
                    validation_split=0.2,
                    verbose=0)
running_time = time() - start_time
```

```
In [187]: plt.plot(history2.history['train hybrid loss'], label = 'train')
          plt.plot(history2.history['val_test hybrid loss'], label = 'validation')
          plt.xlabel('Epoch')
          plt.ylabel('Loss')
          plt.legend(loc='lower right')
          plt.title('Hybrid Loss')
          plt.show()
```

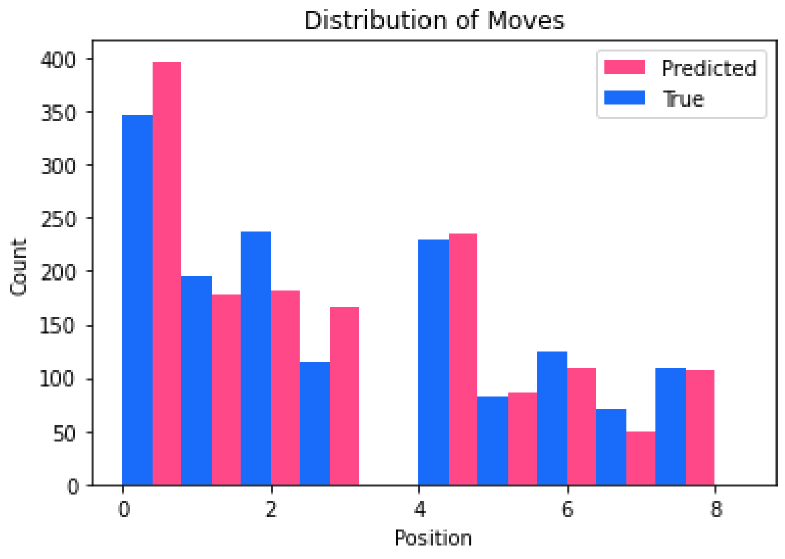


```
In [189]: m_pred, v_pred = model.predict(X_test)
          print(accuracy_score(y_move_test.ravel(), np.argmax(m_pred, axis=1)))
          print(mean_squared_error(y_value_test, v_pred))
```

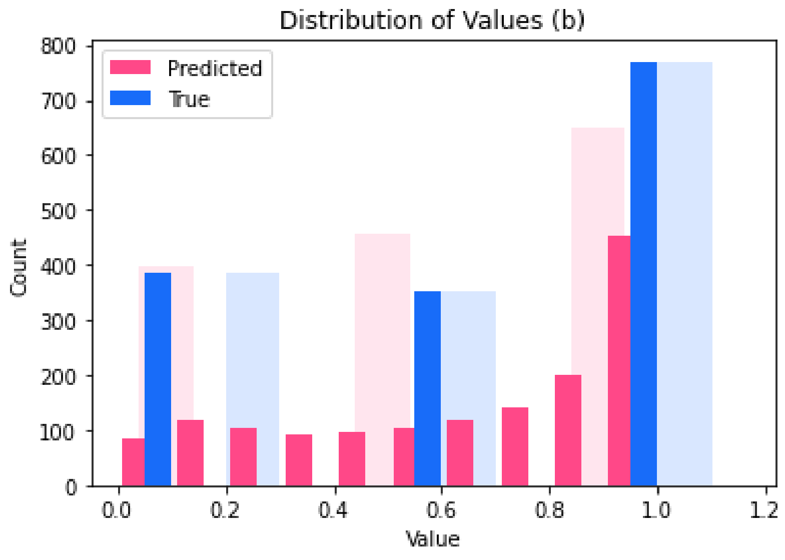
```
0.7803583278035833
0.08021309
```

These values are comparable to the ones when the training was on 100 epochs.

```
In [190]: m_pred_max = [p.tolist().index(max(p)) for p in m_pred]

          plt.hist(m_pred_max,       color='#FF4888', width=0.4, align='right',   label=
          'Predicted')
          plt.hist(y_move_test[:,0], color='#186CF9', width=0.4, align='mid', label='Tru
          e')
          plt.xlabel('Position')
          plt.ylabel('Count')
          plt.title('Distribution of Moves')
          plt.legend()
          plt.show()


          plt.hist((v_pred[:,0],
                     y_value_test[:,0]),
                    width=0.1,
                    align='mid',
                    color=('#FFABC8','#84B2FF'),
                    alpha=0.3,
                    bins=[0,0.4, 0.8, 1.2])

          plt.xlabel('Value')
          plt.ylabel('Count')
          plt.title('Distribution of Values (a)')
          plt.legend()
          #plt.show()

          plt.hist((v_pred[:,0],
                     y_value_test[:,0]),
                    width=0.05,
                    align='mid',
                    color=('#FF4888','#186CF9'),
                    label=('Predicted','True'))
                    #bins=[0,0.2, 0.4,0.6, 0.8,1,1.2])



          plt.xlabel('Value')
          plt.ylabel('Count')
          plt.title('Distribution of Values (b)')
          plt.legend()
          plt.show()
```



```
No handles with labels found to put in legend.
```



The distribution of moves still closely matches the 'true' data, and the distribution of values exhibits a closer match between predicted and actual data.

There are additional ways that we can compare this model against the original, but it is worth exploring, since over-fitting can lead to sub-optimal solutions when looking at unseen data.