INTERNSHIP PROGRESS

Modeling curriculum learning

TABLE OF CONTENTS

WEEKLY OBJECTIVES

	FEBRUARY	MARCH	APRIL	MAY
W1	/	Finish 3 models for Level 1 Level 2: accuracy RL learning	Unify the Level 1 models Account for catastrophic interference	Finishing touches Written report
W2	/	Level 2: accuracy RL Level 2: learning progress RL	Level 3: integrate both accuracy and learning progress	Written report
W3	/	Create hypotheses and model ideas for Level 2 & 3 Prepare the presentation	Level 3: explore more options (chaining effect? Between-task learning?)	(exams)
W4	Level 1: implement tasks and neural network	**LAB PRESENTATION** Integrate all suggestions	Level 3: adjust and compare model performances	(exams)

🟨 Past week

🟩 Upcoming week

LAST WEEK'S OBJECTIVES

1

IMPLEMENT A SIMPLE NEURAL NETWORK

Using the MNIST dataset ✓

LEARN TO TRACK MODEL PARAMETERS

Such as accuracy, loss, performance

2

CREATE THE FIRST TASK

Easy
Make questions ✓

3

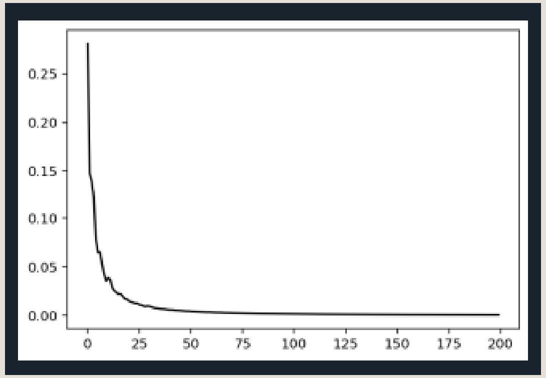
CREATE INTRO FOR THE WEBSITE ✓

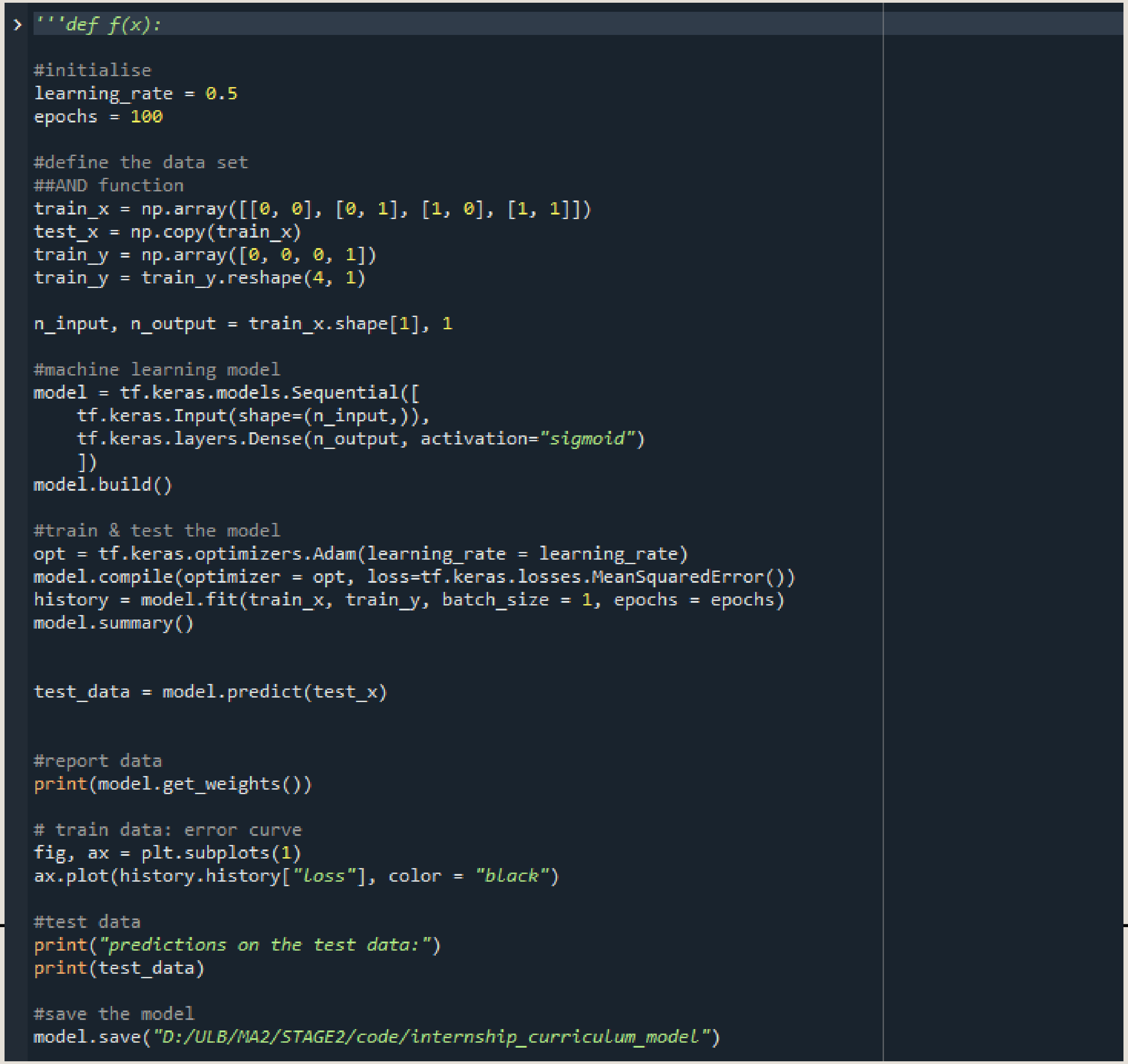
4

SUMMARISE FINDINGS

Write a short summary of findings, questions and plan for next week. ✓

5





```
'''def f(x):

#initialise
learning_rate = 0.5
epochs = 100

#define the data set
##AND function
train_x = np.array([[0, 0], [0, 1], [1, 0], [1, 1]])
test_x = np.copy(train_x)
train_y = np.array([0, 0, 0, 1])
train_y = train_y.reshape(4, 1)

n_input, n_output = train_x.shape[1], 1

#machine learning model
model = tf.keras.models.Sequential([
    tf.keras.Input(shape=(n_input,)),
    tf.keras.layers.Dense(n_output, activation="sigmoid")
    ])
model.build()

#train & test the model
opt = tf.keras.optimizers.Adam(learning_rate = learning_rate)
model.compile(optimizer = opt, loss=tf.keras.losses.MeanSquaredError())
history = model.fit(train_x, train_y, batch_size = 1, epochs = epochs)
model.summary()


test_data = model.predict(test_x)


#report data
print(model.get_weights())

# train data: error curve
fig, ax = plt.subplots(1)
ax.plot(history.history["loss"], color = "black")

#test data
print("predictions on the test data:")
print(test_data)

#save the model
model.save("D:/ULB/MA2/STAGE2/code/internship_curriculum_model")
```

What the code looks like

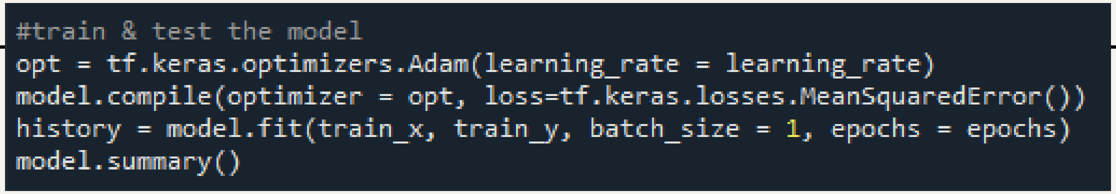
Questions

Question 1

- Why are we using mean squared error and not cross entropy?

➔ assumes linearity in a binary setting, sensitivity to outliers and greater loss penalty

➔ However, sigmoid neuron which accounts for non-linearity?



```
#train & test the model
opt = tf.keras.optimizers.Adam(learning_rate = learning_rate)
model.compile(optimizer = opt, loss=tf.keras.losses.MeanSquaredError())
history = model.fit(train_x, train_y, batch_size = 1, epochs = epochs)
model.summary()
```

Question 2

- Should I add a stop requirement? (Early stopping)

Notes

- XOR is not working yet.

NEXT WEEK'S OBJECTIVES

1

IMPLEMENT ALL 3 TASKS

Easy – Hard – Impossible
Give them the same structure

LEARN TO TRACK MODEL PARAMETERS

Such as accuracy, loss, performance

2

UPLOAD CODE TO GITHUB

Improve understanding of the current model, search for potential improvements.

3

START LEVEL 2: REINFORCEMENT LEARNING

Research teacher level networks, and implement them using accuracy

4

SUMMARISE FINDINGS

Write a short summary of findings, questions and plan for next week.

5