MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE
NATIONAL TECHNICAL UNIVERSITY OF UKRAINE
" IHORY SIKORSKY KYIV POLYTECHNIC INSTITUTE"

Volodymyr Shymkovych

Design and implementation of software systems with neural networks

Design and Implementation of a Neural Network for Binary Classification Using TensorFlow

LABORATORY WORK #1

kulubecioglu mehmet
IM-14 FIOT

Kyiv
IHORY SIKORSKY KYIV POLYTECHNIC INSTITUTE
2024

1. Installing Libraries:

```
In [13]: import tensorflow as tf
         import numpy as np
```

- import tensorflow as tf: Installs the TensorFlow library with the alias tf.
- import numpy as np: Installs the NumPy library with the alias np.

pip install tensorflow

```
In [5]: pip install tensorflow
```

```
7154d678359bed995abdd9daf0cff0/google_auth-2.28.1-py2.py3-none-any.whl.m
etadata
  Using cached google_auth-2.28.1-py2.py3-none-any.whl.metadata (4.7 kB)
Collecting google-auth-oauthlib<2,>=0.5 (from tensorboard<2.16,>=2.15->t
ensorflow-intel==2.15.0->tensorflow)
  Obtaining dependency information for google-auth-oauthlib<2,>=0.5 from
https://files.pythonhosted.org/packages/71/bf/9e125754d1adb3bc4bd206c4e5
df756513b1d23675ac06caa471278d1f3f/google_auth_oauthlib-1.2.0-py2.py3-no
ne-any.whl.metadata
  Using cached google_auth_oauthlib-1.2.0-py2.py3-none-any.whl.metadata
(2.7 kB)
Requirement already satisfied: markdown>=2.6.8 in c:\users\mehme\anacond
a3\lib\site-packages (from tensorboard<2.16,>=2.15->tensorflow-intel==2.
15.0->tensorflow) (3.4.1)
Requirement already satisfied: requests<3,>=2.21.0 in c:\users\mehme\ana
conda3\lib\site-packages (from tensorboard<2.16,>=2.15->tensorflow-intel
==2.15.0->tensorflow) (2.31.0)
Collecting tensorboard-data-server<0.8.0,>=0.7.0 (from tensorboard<2.16,
>=2.15->tensorflow-intel==2.15.0->tensorflow)
  Obtaining dependency information for tensorboard-data-server<0.8.0,>=
```

pip install numpy

```
In [12]: !python -m pip install --upgrade pip

  Downloading pip-24.0-py3-none-any.whl.metadata (3.6 kB)
Downloading pip-24.0-py3-none-any.whl (2.1 MB)
   ---------------------------------------- 0.0/2.1 MB ? eta -:--:--
   ---------------------------------------- 0.0/2.1 MB ? eta -:--:--
   ---------------------------------------- 0.0/2.1 MB ? eta -:--:--
   ---------------------------------------- 0.0/2.1 MB ? eta -:--:--
   ---------------------------------------- 0.0/2.1 MB ? eta -:--:--
   ---------------------------------------- 0.0/2.1 MB ? eta -:--:--
   ---------------------------------------- 0.0/2.1 MB ? eta -:--:--
   ---------------------------------------- 0.0/2.1 MB ? eta -:--:--
   ---------------------------------------- 0.0/2.1 MB ? eta -:--:--
   ---------------------------------------- 0.0/2.1 MB ? eta -:--:--
   ---------------------------------------- 0.0/2.1 MB ? eta -:--:--
    --------------------------------------- 0.0/2.1 MB 640.0 kB/s eta 0:
00:04
    --------------------------------------- 0.0/2.1 MB 640.0 kB/s eta 0:
00:04
    --------------------------------------- 0.0/2.1 MB 640.0 kB/s
00:04
```

2. Identification of Data:

```
# Tanımlama
x = np.array([[0, 0, 0], [0, 1, 1], [1, 0, 1], [1, 1, 0]])
y = np.array([0, 1, 1, 0])
```

- **x:** A NumPy array of values 0 and 1, containing 4 rows and 3 columns.
- **y:** A 4-element NumPy array consisting of values 0 and 1.

3. Model Creation:

```
model = tf.keras.Sequential([
    tf.keras.layers.Dense(3, input_dim=3, activation="tanh"),
    tf.keras.layers.Dense(1, activation="sigmoid")
])
```

- tf.keras.Sequential([...]): Creates a Keras model that defines layers in a sequential manner.
- tf.keras.layers.Dense(3, input_dim=3, activation="tanh"): Creates a hidden layer with 3 neurons and "tanh" activation function.

- tf.keras.layers.Dense(1, activation="sigmoid"): Creates an output layer with 1 neuron and "sigmoid" activation function.

4. Model Compilation:

```
model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=0.05), loss='binary_crossentropy', metrics=['accuracy'])
```

- **model.compile():** Sets the parameters necessary to train the model.
- **optimizer:** Defines the optimization algorithm. Here, the Adam algorithm is selected and the learning rate is set to 0.05.
- **loss:** Defines the loss function. Here the binary cross entropy function is selected.
- **metrics:** Defines the measurements to be tracked during training. Here the accuracy metric is selected.

5. Model Training:

```
# Model eğitme
model.fit(x, y, epochs=100)
```

- **model.fit():** Used to train the model on data.
- **x:** Input data.
- **y:** Target data.
- **epochs:** The number of epochs to be repeated during training. Here, 100 cycles are selected.

6. Model Evaluation:

```
# Model değerlendirme
loss, accuracy = model.evaluate(x, y)
```

- **model.evaluate():** Used to evaluate the performance of the model.
- **x:** Input data.
- **y:** Target data.
- **loss:** Value of the loss function.
- **accuracy:** The value of the accuracy metric.

7. Making Predictions:

```python
# Tahmin yapma
prediction = model.predict(x)
```

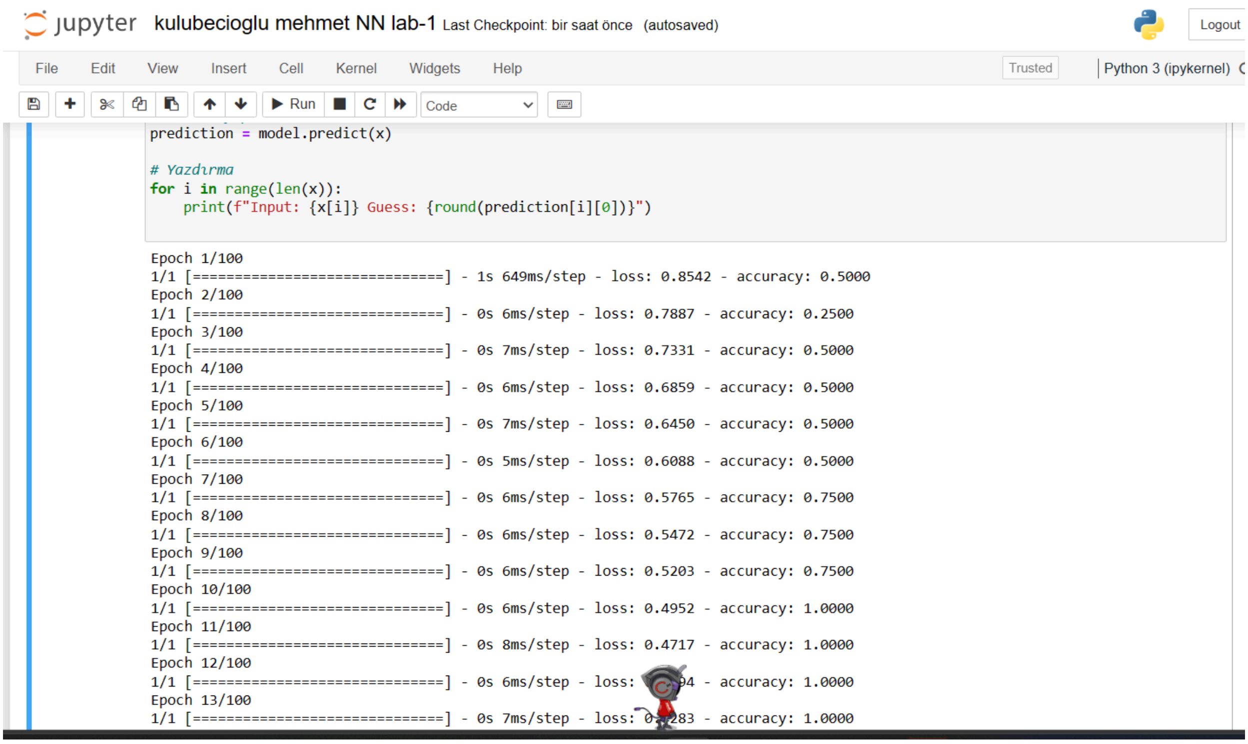
- **model.predict():** Allows the model to make predictions on new data.
- **x:** Input data.
- **prediction:** Values predicted by the model.

8. Printing:

```python
# Yazdırma
for i in range(len(x)):
    print(f"Input: {x[i]} Guess: {round(prediction[i][0])}")
```

- **for loop:** Used to print each input and prediction value.
- **round(prediction[i][0]):** Rounds the prediction value to the nearest integer.

Output:




```
prediction = model.predict(x)

# Yazdırma
for i in range(len(x)):
    print(f"Input: {x[i]} Guess: {round(prediction[i][0])}")
```

```
Epoch 1/100
1/1 [==============================] - 1s 649ms/step - loss: 0.8542 - accuracy: 0.5000
Epoch 2/100
1/1 [==============================] - 0s 6ms/step - loss: 0.7887 - accuracy: 0.2500
Epoch 3/100
1/1 [==============================] - 0s 7ms/step - loss: 0.7331 - accuracy: 0.5000
Epoch 4/100
1/1 [==============================] - 0s 6ms/step - loss: 0.6859 - accuracy: 0.5000
Epoch 5/100
1/1 [==============================] - 0s 7ms/step - loss: 0.6450 - accuracy: 0.5000
Epoch 6/100
1/1 [==============================] - 0s 5ms/step - loss: 0.6088 - accuracy: 0.5000
Epoch 7/100
1/1 [==============================] - 0s 6ms/step - loss: 0.5765 - accuracy: 0.7500
Epoch 8/100
1/1 [==============================] - 0s 6ms/step - loss: 0.5472 - accuracy: 0.7500
Epoch 9/100
1/1 [==============================] - 0s 6ms/step - loss: 0.5203 - accuracy: 0.7500
Epoch 10/100
1/1 [==============================] - 0s 6ms/step - loss: 0.4952 - accuracy: 1.0000
Epoch 11/100
1/1 [==============================] - 0s 8ms/step - loss: 0.4717 - accuracy: 1.0000
Epoch 12/100
1/1 [==============================] - 0s 6ms/step - loss: [illegible]94 - accuracy: 1.0000
Epoch 13/100
1/1 [==============================] - 0s 7ms/step - loss: 0.[illegible]283 - accuracy: 1.0000
```

jupyter kulubecioglu mehmet NN lab-1 Last Checkpoint: bir saat önce (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted

```
1/1 [==============================] - 0s 6ms/step - loss: 0.0101 - accuracy: 1.0000
Epoch 91/100
1/1 [==============================] - 0s 6ms/step - loss: 0.0099 - accuracy: 1.0000
Epoch 92/100
1/1 [==============================] - 0s 7ms/step - loss: 0.0097 - accuracy: 1.0000
Epoch 93/100
1/1 [==============================] - 0s 6ms/step - loss: 0.0095 - accuracy: 1.0000
Epoch 94/100
1/1 [==============================] - 0s 6ms/step - loss: 0.0094 - accuracy: 1.0000
Epoch 95/100
1/1 [==============================] - 0s 5ms/step - loss: 0.0092 - accuracy: 1.0000
Epoch 96/100
1/1 [==============================] - 0s 8ms/step - loss: 0.0090 - accuracy: 1.0000
Epoch 97/100
1/1 [==============================] - 0s 7ms/step - loss: 0.0088 - accuracy: 1.0000
Epoch 98/100
1/1 [==============================] - 0s 7ms/step - loss: 0.0087 - accuracy: 1.0000
Epoch 99/100
1/1 [==============================] - 0s 5ms/step - loss: 0.0085 - accuracy: 1.0000
Epoch 100/100
1/1 [==============================] - 0s 6ms/step - loss: 0.0084 - accuracy: 1.0000
1/1 [==============================] - 0s 151ms/step - loss: 0.0082 - accuracy: 1.0000
1/1 [==============================] - 0s 58ms/step
Input: [0 0 0] Guess: 0
Input: [0 1 1] Guess: 1
Input: [1 0 1] Guess: 1
Input: [1 1 0] Guess: 0
```

In [ ]: