STAT 6340 Bonus Project

Lakshmipriya Narayanan

SECTION 1: Observations and Answers

1

(a): See code attached in Section 2.

(b):

Layer	**Input**	**Output**	**Number of parameters in each layer**
C1	(28, 28, 1)	(28 - 2, 28 - 2, 32 filters) = **(26, 26, 32)**	32 x (3 x 3 + 1) = **320**
P1	(26, 26, 32)	($\frac{26}{2}$, $\frac{26}{2}$, 32) = **(13, 13, 32)**	-
C2	(13, 13, 32)	(13 – 2, 13 – 2, 64 filters) = **(11, 11, 64)**	64 x (32 x [3 x 3] + 1) = **18,496**
P2	(11, 11, 64)	($\frac{11}{2}$, $\frac{11}{2}$, 64) = **(5, 5, 64)**	-
C3	(5, 5, 64)	(5 – 2, 5 – 2, 64 filters) = **(3, 3, 64)**	64 x (64 x [3 x 3] + 1) = **36, 928**
F1	(3, 3, 64)	(3 x 3 x 64, 1) = **(576, 1)**	-
D	(576, 1)	(size = 64 , 1) = **(64, 1)**	64 x (576 + 1) = **36,928**
D1	(64, 1)	**(10, 1)**	10 x (64 + 1) = **650**
Total number of parameters = 320 + 18,496 + 36,928 + 36,928 + 650 = **93,322**			

(c): Test accuracy = 98.07 %

2

Number of layers = 3
Mini batch size = 512
Learning rate = 0.01

Hidden units	**Dropout**	**Test Accuracy**	**Test Loss**
16	0%	0.854	1.325
32	0%	0.852	1.554
64	0%	0.865	1.186
16	30%	0.869	1.301
32	30%	0.868	0.854
64	30%	0.866	0.841

From the above table we observe that without 30% dropout and increasing the number of hidden units from 16 to 64 improved test accuracy. For 30% dropout, the least number of hidden units (16) had the highest test accuracy. Generally, we can say that adding dropout regularization improved test accuracy and the most significant improvement was seen with 32 hidden units where the test loss substantially decreased from 1.554 to 0.854.

In conclusion, increasing the number of hidden units and adding dropout regularization improves the model performance.

SECTION 2: CODE

Problem 1:

```
#!pip install tensorflow

#import tensorflow as tf
#print(tf.__version__)

# Import required libraries
import numpy as np
import tensorflow as tf
import random
from keras.datasets import mnist
from keras.models import Sequential
from keras.layers import Dense, Flatten, Dropout, Conv2D, MaxPooling2D
from keras.utils import to_categorical
from keras.optimizers import RMSprop

# Load the MNIST dataset from keras.datasets and assign training and
test data
# x_train and x_test contain the images while y_train and y_test
contain digits 0-9
(x_train, y_train), (x_test, y_test) = mnist.load_data()

# Scale the images(in pixels to floating point) to a range of 0-1 by
dividing
# by 255 since 0-255 is the initial range. This normalization improves
the performance
# and convergence
x_train = x_train.astype('float32') / 255
x_test = x_test.astype('float32') / 255

# Convert categorical labels into binary matrix (one-hot encoding)
because the
# labels are initially integers from 0-9
y_train = to_categorical(y_train, 10)
y_test = to_categorical(y_test, 10)


# Reshape input data to include channel dimension for CNN
x_train = x_train.reshape(x_train.shape[0], 28, 28, 1) #reshaped from
(60000, 28, 28)
x_test = x_test.reshape(x_test.shape[0], 28, 28, 1) #reshaped from
(10000, 28, 28)

# Set random seeds for reproducibility
np.random.seed(123)
tf.random.set_seed(123)

# Build the CNN model with the specified architecture
model = Sequential([
    # First convolution layer with 32 3x3 filters, followed by max
pooling.
```

```
    # Activation function is ReLU. 28 x 28 pixels and 1 color channel
is the input layer.
    Conv2D(32, (3, 3), activation = 'relu', input_shape = (28, 28,
1)),
    #First maxpooling layer
    MaxPooling2D((2, 2)),
    # Second convolution layer with 64 3x3 filters
    Conv2D(64, (3, 3), activation = 'relu'),
    #Seconf maxpoolin layer
    MaxPooling2D((2, 2)),
    # Third convolution layer with 64 3x3 filters.
    # Shape of this output layer will be (3,3,64)
    Conv2D(64, (3, 3), activation = 'relu'),
    # Flatten the output
    Flatten(),
    # One Dense layer with 64 units
    Dense(64, activation = 'relu'),
    # Output layer
    Dense(10, activation = 'softmax')
])

# Compile the model with RMSprop optimizer and learning rate = 0.01
model.compile(optimizer = RMSprop(learning_rate = 0.01), loss =
'categorical_crossentropy', metrics = ['accuracy'])

# Print model summary to verify the architecture specified in (a)
model.summary()

# Train the model on the training data with specified mini batch size
and epochs
history = model.fit(x_train, y_train, batch_size = 64, epochs = 5,
validation_split = 0.1, verbose = 1)

# Evaluate the model on test data
test_loss, test_acc = model.evaluate(x_test, y_test, verbose = 1)
print(f'\nTest accuracy: {test_acc * 100:.6f}%')
```

```
Model: "sequential_1"

┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━
━━━━━┓
┃ Layer (type)                    ┃ Output Shape           ┃
Param # ┃
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━
━━━━━┩
│ conv2d_3 (Conv2D)               │ (None, 26, 26, 32)     │
320 │
├─────────────────────────────────┼────────────────────────┼──────────
─────┤
│ max_pooling2d_2 (MaxPooling2D)  │ (None, 13, 13, 32)     │
```

```
0 │
├─────────────────────────────────┼────────────────────────┼──────────
─────┤
│ conv2d_4 (Conv2D)               │ (None, 11, 11, 64)     │
18,496 │
├─────────────────────────────────┼────────────────────────┼──────────
─────┤
│ max_pooling2d_3 (MaxPooling2D)  │ (None, 5, 5, 64)       │
0 │
├─────────────────────────────────┼────────────────────────┼──────────
─────┤
│ conv2d_5 (Conv2D)               │ (None, 3, 3, 64)       │
36,928 │
├─────────────────────────────────┼────────────────────────┼──────────
─────┤
│ flatten_1 (Flatten)             │ (None, 576)            │
0 │
├─────────────────────────────────┼────────────────────────┼──────────
─────┤
│ dense_2 (Dense)                 │ (None, 64)             │
36,928 │
├─────────────────────────────────┼────────────────────────┼──────────
─────┤
│ dense_3 (Dense)                 │ (None, 10)             │
650 │
└─────────────────────────────────┴────────────────────────┴──────────
─────┘

 Total params: 93,322 (364.54 KB)

 Trainable params: 93,322 (364.54 KB)

 Non-trainable params: 0 (0.00 B)

Epoch 1/5
844/844 ━━━━━━━━━━━━━━━━━━━━ 34s 38ms/step - accuracy: 0.7926 - loss:
0.6455 - val_accuracy: 0.9828 - val_loss: 0.0602
Epoch 2/5
844/844 ━━━━━━━━━━━━━━━━━━━━ 33s 39ms/step - accuracy: 0.9754 - loss:
0.0901 - val_accuracy: 0.9860 - val_loss: 0.0630
Epoch 3/5
844/844 ━━━━━━━━━━━━━━━━━━━━ 33s 39ms/step - accuracy: 0.9787 - loss:
0.0819 - val_accuracy: 0.9843 - val_loss: 0.0724
Epoch 4/5
844/844 ━━━━━━━━━━━━━━━━━━━━ 33s 39ms/step - accuracy: 0.9790 - loss:
0.0828 - val_accuracy: 0.9802 - val_loss: 0.0837
Epoch 5/5
844/844 ━━━━━━━━━━━━━━━━━━━━ 32s 37ms/step - accuracy: 0.9806 - loss:
0.0751 - val_accuracy: 0.9793 - val_loss: 0.0930
313/313 ━━━━━━━━━━━━━━━━━━━━ 4s 13ms/step - accuracy: 0.9779 - loss:
```

```
0.0981

Test accuracy: 98.070002%
```

(2): Repeat the analysis of Lab 10.9.5 on the IMDb data using a similarly structured neural network. We used 16 hidden units at each of two hidden layers. Explore the effect of increasing this to 32 and 64 units per layer, with and without 30% dropout regularization.

```
!pip install ISLP
```

```
Collecting ISLP
  Downloading ISLP-0.4.0-py3-none-any.whl.metadata (7.0 kB)
Requirement already satisfied: numpy>=1.7.1 in
/usr/local/lib/python3.10/dist-packages (from ISLP) (1.26.4)
Requirement already satisfied: scipy>=0.9 in
/usr/local/lib/python3.10/dist-packages (from ISLP) (1.13.1)
Requirement already satisfied: pandas>=0.20 in
/usr/local/lib/python3.10/dist-packages (from ISLP) (2.2.2)
Requirement already satisfied: lxml in /usr/local/lib/python3.10/dist-
packages (from ISLP) (5.3.0)
Requirement already satisfied: scikit-learn>=1.2 in
/usr/local/lib/python3.10/dist-packages (from ISLP) (1.5.2)
Requirement already satisfied: joblib in
/usr/local/lib/python3.10/dist-packages (from ISLP) (1.4.2)
Requirement already satisfied: statsmodels>=0.13 in
/usr/local/lib/python3.10/dist-packages (from ISLP) (0.14.4)
Collecting lifelines (from ISLP)
  Downloading lifelines-0.30.0-py3-none-any.whl.metadata (3.2 kB)
Collecting pygam (from ISLP)
  Downloading pygam-0.9.1-py3-none-any.whl.metadata (7.1 kB)
Requirement already satisfied: torch in
/usr/local/lib/python3.10/dist-packages (from ISLP) (2.5.1+cu121)
Collecting pytorch-lightning (from ISLP)
  Downloading pytorch_lightning-2.4.0-py3-none-any.whl.metadata (21
kB)
Collecting torchmetrics (from ISLP)
  Downloading torchmetrics-1.6.0-py3-none-any.whl.metadata (20 kB)
Requirement already satisfied: python-dateutil>=2.8.2 in
/usr/local/lib/python3.10/dist-packages (from pandas>=0.20->ISLP)
(2.8.2)
Requirement already satisfied: pytz>=2020.1 in
/usr/local/lib/python3.10/dist-packages (from pandas>=0.20->ISLP)
(2024.2)
Requirement already satisfied: tzdata>=2022.7 in
/usr/local/lib/python3.10/dist-packages (from pandas>=0.20->ISLP)
(2024.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in
/usr/local/lib/python3.10/dist-packages (from scikit-learn>=1.2->ISLP)
(3.5.0)
Requirement already satisfied: patsy>=0.5.6 in
/usr/local/lib/python3.10/dist-packages (from statsmodels>=0.13->ISLP)
(1.0.1)
Requirement already satisfied: packaging>=21.3 in
```

```
/usr/local/lib/python3.10/dist-packages (from statsmodels>=0.13->ISLP)
(24.2)
Requirement already satisfied: matplotlib>=3.0 in
/usr/local/lib/python3.10/dist-packages (from lifelines->ISLP) (3.8.0)
Requirement already satisfied: autograd>=1.5 in
/usr/local/lib/python3.10/dist-packages (from lifelines->ISLP) (1.7.0)
Collecting autograd-gamma>=0.3 (from lifelines->ISLP)
  Downloading autograd-gamma-0.5.0.tar.gz (4.0 kB)
  Preparing metadata (setup.py) ... ulaic>=0.2.2 (from lifelines-
>ISLP)
  Downloading formulaic-1.0.2-py3-none-any.whl.metadata (6.8 kB)
Requirement already satisfied: progressbar2<5.0.0,>=4.2.0 in
/usr/local/lib/python3.10/dist-packages (from pygam->ISLP) (4.5.0)
Collecting scipy>=0.9 (from ISLP)
  Downloading scipy-1.11.4-cp310-cp310-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (60 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 60.4/60.4 kB 2.7 MB/s eta
0:00:00
ent already satisfied: tqdm>=4.57.0 in /usr/local/lib/python3.10/dist-
packages (from pytorch-lightning->ISLP) (4.66.6)
Requirement already satisfied: PyYAML>=5.4 in
/usr/local/lib/python3.10/dist-packages (from pytorch-lightning->ISLP)
(6.0.2)
Requirement already satisfied: fsspec>=2022.5.0 in
/usr/local/lib/python3.10/dist-packages (from fsspec[http]>=2022.5.0-
>pytorch-lightning->ISLP) (2024.10.0)
Requirement already satisfied: typing-extensions>=4.4.0 in
/usr/local/lib/python3.10/dist-packages (from pytorch-lightning->ISLP)
(4.12.2)
Collecting lightning-utilities>=0.10.0 (from pytorch-lightning->ISLP)
  Downloading lightning_utilities-0.11.9-py3-none-any.whl.metadata
(5.2 kB)
Requirement already satisfied: filelock in
/usr/local/lib/python3.10/dist-packages (from torch->ISLP) (3.16.1)
Requirement already satisfied: networkx in
/usr/local/lib/python3.10/dist-packages (from torch->ISLP) (3.4.2)
Requirement already satisfied: jinja2 in
/usr/local/lib/python3.10/dist-packages (from torch->ISLP) (3.1.4)
Requirement already satisfied: sympy==1.13.1 in
/usr/local/lib/python3.10/dist-packages (from torch->ISLP) (1.13.1)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in
/usr/local/lib/python3.10/dist-packages (from sympy==1.13.1->torch-
>ISLP) (1.3.0)
Collecting interface-meta>=1.2.0 (from formulaic>=0.2.2->lifelines-
>ISLP)
  Downloading interface_meta-1.3.0-py3-none-any.whl.metadata (6.7 kB)
Requirement already satisfied: wrapt>=1.0 in
/usr/local/lib/python3.10/dist-packages (from formulaic>=0.2.2-
>lifelines->ISLP) (1.17.0)
```

```
Requirement already satisfied: aiohttp!=4.0.0a0,!=4.0.0a1 in
/usr/local/lib/python3.10/dist-packages (from fsspec[http]>=2022.5.0-
>pytorch-lightning->ISLP) (3.11.9)
Requirement already satisfied: setuptools in
/usr/local/lib/python3.10/dist-packages (from lightning-
utilities>=0.10.0->pytorch-lightning->ISLP) (75.1.0)
Requirement already satisfied: contourpy>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib>=3.0-
>lifelines->ISLP) (1.3.1)
Requirement already satisfied: cycler>=0.10 in
/usr/local/lib/python3.10/dist-packages (from matplotlib>=3.0-
>lifelines->ISLP) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib>=3.0-
>lifelines->ISLP) (4.55.1)
Requirement already satisfied: kiwisolver>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib>=3.0-
>lifelines->ISLP) (1.4.7)
Requirement already satisfied: pillow>=6.2.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib>=3.0-
>lifelines->ISLP) (11.0.0)
Requirement already satisfied: pyparsing>=2.3.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib>=3.0-
>lifelines->ISLP) (3.2.0)
Requirement already satisfied: python-utils>=3.8.1 in
/usr/local/lib/python3.10/dist-packages (from
progressbar2<5.0.0,>=4.2.0->pygam->ISLP) (3.9.1)
Requirement already satisfied: six>=1.5 in
/usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2-
>pandas>=0.20->ISLP) (1.16.0)
Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.10/dist-packages (from jinja2->torch->ISLP)
(3.0.2)
Requirement already satisfied: aiohappyeyeballs>=2.3.0 in
/usr/local/lib/python3.10/dist-packages (from aiohttp!=4.0.0a0,!
=4.0.0a1->fsspec[http]>=2022.5.0->pytorch-lightning->ISLP) (2.4.4)
Requirement already satisfied: aiosignal>=1.1.2 in
/usr/local/lib/python3.10/dist-packages (from aiohttp!=4.0.0a0,!
=4.0.0a1->fsspec[http]>=2022.5.0->pytorch-lightning->ISLP) (1.3.1)
Requirement already satisfied: async-timeout<6.0,>=4.0 in
/usr/local/lib/python3.10/dist-packages (from aiohttp!=4.0.0a0,!
=4.0.0a1->fsspec[http]>=2022.5.0->pytorch-lightning->ISLP) (4.0.3)
Requirement already satisfied: attrs>=17.3.0 in
/usr/local/lib/python3.10/dist-packages (from aiohttp!=4.0.0a0,!
=4.0.0a1->fsspec[http]>=2022.5.0->pytorch-lightning->ISLP) (24.2.0)
Requirement already satisfied: frozenlist>=1.1.1 in
/usr/local/lib/python3.10/dist-packages (from aiohttp!=4.0.0a0,!
=4.0.0a1->fsspec[http]>=2022.5.0->pytorch-lightning->ISLP) (1.5.0)
Requirement already satisfied: multidict<7.0,>=4.5 in
```

```
/usr/local/lib/python3.10/dist-packages (from aiohttp!=4.0.0a0,!
=4.0.0a1->fsspec[http]>=2022.5.0->pytorch-lightning->ISLP) (6.1.0)
Requirement already satisfied: propcache>=0.2.0 in
/usr/local/lib/python3.10/dist-packages (from aiohttp!=4.0.0a0,!
=4.0.0a1->fsspec[http]>=2022.5.0->pytorch-lightning->ISLP) (0.2.1)
Requirement already satisfied: yarl<2.0,>=1.17.0 in
/usr/local/lib/python3.10/dist-packages (from aiohttp!=4.0.0a0,!
=4.0.0a1->fsspec[http]>=2022.5.0->pytorch-lightning->ISLP) (1.18.3)
Requirement already satisfied: idna>=2.0 in
/usr/local/lib/python3.10/dist-packages (from yarl<2.0,>=1.17.0-
>aiohttp!=4.0.0a0,!=4.0.0a1->fsspec[http]>=2022.5.0->pytorch-
lightning->ISLP) (3.10)
Downloading ISLP-0.4.0-py3-none-any.whl (3.6 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3.6/3.6 MB 35.4 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 349.3/349.3 kB 27.9 MB/s eta
0:00:00
-0.9.1-py3-none-any.whl (522 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 522.0/522.0 kB 34.3 MB/s eta
0:00:00
anylinux_2_17_x86_64.manylinux2014_x86_64.whl (36.4 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 36.4/36.4 MB 33.2 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 815.2/815.2 kB 36.5 MB/s eta
0:00:00
etrics-1.6.0-py3-none-any.whl (926 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 926.4/926.4 kB 42.8 MB/s eta
0:00:00
ulaic-1.0.2-py3-none-any.whl (94 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 94.5/94.5 kB 8.4 MB/s eta
0:00:00
eta-1.3.0-py3-none-any.whl (14 kB)
Building wheels for collected packages: autograd-gamma
  Building wheel for autograd-gamma (setup.py) ... ma:
filename=autograd_gamma-0.5.0-py3-none-any.whl size=4031
sha256=0c377581eb67a5072de8b70d5e3f4a10e3ae74d71d5021f6c2b5b5cc1f33da5
a
  Stored in directory:
/root/.cache/pip/wheels/25/cc/e0/ef2969164144c899fedb22b338f6703e2b9cf
46eeebf254991
Successfully built autograd-gamma
Installing collected packages: scipy, lightning-utilities, interface-
meta, autograd-gamma, torchmetrics, pygam, formulaic, lifelines,
pytorch-lightning, ISLP
  Attempting uninstall: scipy
    Found existing installation: scipy 1.13.1
    Uninstalling scipy-1.13.1:
      Successfully uninstalled scipy-1.13.1
Successfully installed ISLP-0.4.0 autograd-gamma-0.5.0 formulaic-1.0.2
```

```
interface-meta-1.3.0 lifelines-0.30.0 lightning-utilities-0.11.9
pygam-0.9.1 pytorch-lightning-2.4.0 scipy-1.11.4 torchmetrics-1.6.0
```

```
!pip uninstall sympy -y
!pip install sympy
```

```
Found existing installation: sympy 1.13.1
Uninstalling sympy-1.13.1:
  Successfully uninstalled sympy-1.13.1
Collecting sympy
  Downloading sympy-1.13.3-py3-none-any.whl.metadata (12 kB)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in
/usr/local/lib/python3.10/dist-packages (from sympy) (1.3.0)
Downloading sympy-1.13.3-py3-none-any.whl (6.2 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 6.2/6.2 MB 37.6 MB/s eta
0:00:00
py
ERROR: pip's dependency resolver does not currently take into account
all the packages that are installed. This behaviour is the source of
the following dependency conflicts.
torch 2.5.1+cu121 requires sympy==1.13.1; python_version >= "3.9", but
you have sympy 1.13.3 which is incompatible.
Successfully installed sympy-1.13.3
```

```
!pip install torchinfo
!pip uninstall pytorch-lightning
!pip install pytorch-lightning
!pip install torchvision
```

```
Requirement already satisfied: torchinfo in
/usr/local/lib/python3.10/dist-packages (1.8.0)
Found existing installation: pytorch-lightning 2.4.0
Uninstalling pytorch-lightning-2.4.0:
  Would remove:
    /usr/local/lib/python3.10/dist-packages/lightning_fabric/*
    /usr/local/lib/python3.10/dist-packages/pytorch_lightning-
2.4.0.dist-info/*
    /usr/local/lib/python3.10/dist-packages/pytorch_lightning/*
Proceed (Y/n)? Y
  Successfully uninstalled pytorch-lightning-2.4.0
Collecting pytorch-lightning
  Using cached pytorch_lightning-2.4.0-py3-none-any.whl.metadata (21
kB)
Requirement already satisfied: torch>=2.1.0 in
/usr/local/lib/python3.10/dist-packages (from pytorch-lightning)
(2.5.1+cu121)
Requirement already satisfied: tqdm>=4.57.0 in
/usr/local/lib/python3.10/dist-packages (from pytorch-lightning)
(4.66.6)
Requirement already satisfied: PyYAML>=5.4 in
```

```
/usr/local/lib/python3.10/dist-packages (from pytorch-lightning)
(6.0.2)
Requirement already satisfied: fsspec>=2022.5.0 in
/usr/local/lib/python3.10/dist-packages (from fsspec[http]>=2022.5.0-
>pytorch-lightning) (2024.10.0)
Requirement already satisfied: torchmetrics>=0.7.0 in
/usr/local/lib/python3.10/dist-packages (from pytorch-lightning)
(1.6.0)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.10/dist-packages (from pytorch-lightning)
(24.2)
Requirement already satisfied: typing-extensions>=4.4.0 in
/usr/local/lib/python3.10/dist-packages (from pytorch-lightning)
(4.12.2)
Requirement already satisfied: lightning-utilities>=0.10.0 in
/usr/local/lib/python3.10/dist-packages (from pytorch-lightning)
(0.11.9)
Requirement already satisfied: aiohttp!=4.0.0a0,!=4.0.0a1 in
/usr/local/lib/python3.10/dist-packages (from fsspec[http]>=2022.5.0-
>pytorch-lightning) (3.11.9)
Requirement already satisfied: setuptools in
/usr/local/lib/python3.10/dist-packages (from lightning-
utilities>=0.10.0->pytorch-lightning) (75.1.0)
Requirement already satisfied: filelock in
/usr/local/lib/python3.10/dist-packages (from torch>=2.1.0->pytorch-
lightning) (3.16.1)
Requirement already satisfied: networkx in
/usr/local/lib/python3.10/dist-packages (from torch>=2.1.0->pytorch-
lightning) (3.4.2)
Requirement already satisfied: jinja2 in
/usr/local/lib/python3.10/dist-packages (from torch>=2.1.0->pytorch-
lightning) (3.1.4)
Collecting sympy==1.13.1 (from torch>=2.1.0->pytorch-lightning)
  Downloading sympy-1.13.1-py3-none-any.whl.metadata (12 kB)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in
/usr/local/lib/python3.10/dist-packages (from sympy==1.13.1-
>torch>=2.1.0->pytorch-lightning) (1.3.0)
Requirement already satisfied: numpy>1.20.0 in
/usr/local/lib/python3.10/dist-packages (from torchmetrics>=0.7.0-
>pytorch-lightning) (1.26.4)
Requirement already satisfied: aiohappyeyeballs>=2.3.0 in
/usr/local/lib/python3.10/dist-packages (from aiohttp!=4.0.0a0,!
=4.0.0a1->fsspec[http]>=2022.5.0->pytorch-lightning) (2.4.4)
Requirement already satisfied: aiosignal>=1.1.2 in
/usr/local/lib/python3.10/dist-packages (from aiohttp!=4.0.0a0,!
=4.0.0a1->fsspec[http]>=2022.5.0->pytorch-lightning) (1.3.1)
Requirement already satisfied: async-timeout<6.0,>=4.0 in
/usr/local/lib/python3.10/dist-packages (from aiohttp!=4.0.0a0,!
=4.0.0a1->fsspec[http]>=2022.5.0->pytorch-lightning) (4.0.3)
```

```
Requirement already satisfied: attrs>=17.3.0 in
/usr/local/lib/python3.10/dist-packages (from aiohttp!=4.0.0a0,!
=4.0.0a1->fsspec[http]>=2022.5.0->pytorch-lightning) (24.2.0)
Requirement already satisfied: frozenlist>=1.1.1 in
/usr/local/lib/python3.10/dist-packages (from aiohttp!=4.0.0a0,!
=4.0.0a1->fsspec[http]>=2022.5.0->pytorch-lightning) (1.5.0)
Requirement already satisfied: multidict<7.0,>=4.5 in
/usr/local/lib/python3.10/dist-packages (from aiohttp!=4.0.0a0,!
=4.0.0a1->fsspec[http]>=2022.5.0->pytorch-lightning) (6.1.0)
Requirement already satisfied: propcache>=0.2.0 in
/usr/local/lib/python3.10/dist-packages (from aiohttp!=4.0.0a0,!
=4.0.0a1->fsspec[http]>=2022.5.0->pytorch-lightning) (0.2.1)
Requirement already satisfied: yarl<2.0,>=1.17.0 in
/usr/local/lib/python3.10/dist-packages (from aiohttp!=4.0.0a0,!
=4.0.0a1->fsspec[http]>=2022.5.0->pytorch-lightning) (1.18.3)
Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.10/dist-packages (from jinja2->torch>=2.1.0-
>pytorch-lightning) (3.0.2)
Requirement already satisfied: idna>=2.0 in
/usr/local/lib/python3.10/dist-packages (from yarl<2.0,>=1.17.0-
>aiohttp!=4.0.0a0,!=4.0.0a1->fsspec[http]>=2022.5.0->pytorch-
lightning) (3.10)
Using cached pytorch_lightning-2.4.0-py3-none-any.whl (815 kB)
Downloading sympy-1.13.1-py3-none-any.whl (6.2 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 6.2/6.2 MB 34.1 MB/s eta
0:00:00
py, pytorch-lightning
  Attempting uninstall: sympy
    Found existing installation: sympy 1.13.3
    Uninstalling sympy-1.13.3:
      Successfully uninstalled sympy-1.13.3
Successfully installed pytorch-lightning-2.4.0 sympy-1.13.1
Requirement already satisfied: torchvision in
/usr/local/lib/python3.10/dist-packages (0.20.1+cu121)
Requirement already satisfied: numpy in
/usr/local/lib/python3.10/dist-packages (from torchvision) (1.26.4)
Requirement already satisfied: torch==2.5.1 in
/usr/local/lib/python3.10/dist-packages (from torchvision)
(2.5.1+cu121)
Requirement already satisfied: pillow!=8.3.*,>=5.3.0 in
/usr/local/lib/python3.10/dist-packages (from torchvision) (11.0.0)
Requirement already satisfied: filelock in
/usr/local/lib/python3.10/dist-packages (from torch==2.5.1-
>torchvision) (3.16.1)
Requirement already satisfied: typing-extensions>=4.8.0 in
/usr/local/lib/python3.10/dist-packages (from torch==2.5.1-
>torchvision) (4.12.2)
Requirement already satisfied: networkx in
/usr/local/lib/python3.10/dist-packages (from torch==2.5.1-
>torchvision) (3.4.2)
Requirement already satisfied: jinja2 in
/usr/local/lib/python3.10/dist-packages (from torch==2.5.1-
>torchvision) (3.1.4)
Requirement already satisfied: fsspec in
/usr/local/lib/python3.10/dist-packages (from torch==2.5.1-
>torchvision) (2024.10.0)
Requirement already satisfied: sympy==1.13.1 in
/usr/local/lib/python3.10/dist-packages (from torch==2.5.1-
>torchvision) (1.13.1)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in
/usr/local/lib/python3.10/dist-packages (from sympy==1.13.1-
>torch==2.5.1->torchvision) (1.3.0)
Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.10/dist-packages (from jinja2->torch==2.5.1-
>torchvision) (3.0.2)
```

```python
# Import necessary libraries
import numpy as np, pandas as pd
from matplotlib.pyplot import subplots
from sklearn.linear_model import (LinearRegression,
LogisticRegression, Lasso)
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import KFold
from sklearn.pipeline import Pipeline

import ISLP
from ISLP import load_data
from ISLP.models import ModelSpec as MS
from sklearn.model_selection import (train_test_split, GridSearchCV)
import tensorflow as tf
import random
from keras.datasets import mnist
from keras.models import Sequential
from keras.layers import Dense, Flatten, Dropout, Conv2D, MaxPooling2D
from keras.utils import to_categorical
from keras.optimizers import RMSprop

import torch
#from torch.nn import nn
from torch.optim import RMSprop
from torch.utils.data import TensorDataset
from torchmetrics import (MeanAbsoluteError, R2Score)
from torchinfo import summary
from torchvision.io import read_image
from pytorch_lightning import Trainer
from pytorch_lightning.loggers import CSVLogger
from torchvision.datasets import MNIST, CIFAR100
from torchvision.models import (resnet50, ResNet50_Weights)
from torchvision.transforms import (Resize, Normalize, CenterCrop,
ToTensor)

from ISLP.torch import (SimpleDataModule, SimpleModule, ErrorTracker,
rec_num_workers)
from ISLP.torch.imdb import (load_lookup, load_tensor, load_sparse,
load_sequential)
```

FOLLOWING CODE SECTIONS COPY PASTED FROM LAB 10.9.5 IN THE TEXTBOOK

```
(imdb_seq_train, imdb_seq_test) = load_sequential(root='data/IMDB')
padded_sample = np.asarray(imdb_seq_train.tensors[0][0])
sample_review = padded_sample[padded_sample > 0][:12]
sample_review[:12]

lookup = load_lookup(root='data/IMDB')
' '.join(lookup[i] for i in sample_review)
max_num_workers=10
(imdb_train, imdb_test) = load_tensor(root='data/IMDB')
imdb_dm = SimpleDataModule(imdb_train, imdb_test, validation=2000,
num_workers=min(6, max_num_workers), batch_size=512)

class IMDBModel(nn.Module):
    def __init__(self, input_size):
        super(IMDBModel, self).__init__()
        self.dense1 = nn.Linear(input_size, 16)
        self.activation = nn.ReLU()
        self.dense2 = nn.Linear(16, 16)
        self.output = nn.Linear(16, 1)

    # The forward function should be at the same level as __init__
    def forward(self, x):
        val = x
        for _map in [self.dense1, self.activation, self.dense2,
self.activation, self.output]:
            val = _map(val)
        return torch.flatten(val)

imdb_model = IMDBModel(imdb_test.tensors[0].size()[1])
summary(imdb_model, input_size=imdb_test.tensors[0].size(),
col_names=['input_size', 'output_size', 'num_params'])

imdb_optimizer = RMSprop(imdb_model.parameters(), lr=0.001)
imdb_module = SimpleModule.binary_classification(imdb_model,
optimizer=imdb_optimizer)
imdb_logger = CSVLogger('logs', name='IMDB')
imdb_trainer = Trainer(deterministic=True, max_epochs=30,
logger=imdb_logger, callbacks=[ErrorTracker()])
imdb_trainer.fit(imdb_module, datamodule=imdb_dm)
test_res = imdb_trainer.test(imdb_module, datamodule=imdb_dm)
test_res
```

```
/usr/local/lib/python3.10/dist-packages/ISLP/torch/imdb.py:131:
FutureWarning: You are using `torch.load` with `weights_only=False`
(the current default value), which uses the default pickle module
implicitly. It is possible to construct malicious pickle data which
will execute arbitrary code during unpickling (See
https://github.com/pytorch/pytorch/blob/main/SECURITY.md#untrusted-
models for more details). In a future release, the default value for
`weights_only` will be flipped to `True`. This limits the functions
that could be executed during unpickling. Arbitrary objects will no
longer be allowed to be loaded via this mode unless they are
explicitly allowlisted by the user via
`torch.serialization.add_safe_globals`. We recommend you start setting
`weights_only=True` for any use case where you don't have full control
of the loaded file. Please open an issue on GitHub for any issues
related to this experimental feature.
  S_test) = [torch.load(_get_imdb(f'IMDB_{r}', root))
/usr/local/lib/python3.10/dist-packages/ISLP/torch/imdb.py:113:
FutureWarning: You are using `torch.load` with `weights_only=False`
(the current default value), which uses the default pickle module
implicitly. It is possible to construct malicious pickle data which
will execute arbitrary code during unpickling (See
https://github.com/pytorch/pytorch/blob/main/SECURITY.md#untrusted-
models for more details). In a future release, the default value for
`weights_only` will be flipped to `True`. This limits the functions
that could be executed during unpickling. Arbitrary objects will no
longer be allowed to be loaded via this mode unless they are
explicitly allowlisted by the user via
`torch.serialization.add_safe_globals`. We recommend you start setting
`weights_only=True` for any use case where you don't have full control
of the loaded file. Please open an issue on GitHub for any issues
related to this experimental feature.
  X_test, X_train = [torch.load(_get_imdb(f'IMDB_{r}', root))
INFO:pytorch_lightning.utilities.rank_zero:GPU available: False, used:
False
INFO:pytorch_lightning.utilities.rank_zero:TPU available: False,
using: 0 TPU cores
INFO:pytorch_lightning.utilities.rank_zero:HPU available: False,
using: 0 HPUs
INFO:pytorch_lightning.callbacks.model_summary:
  | Name  | Type              | Params | Mode 
----------------------------------------------------
0 | model | IMDBModel         | 160 K  | train
1 | loss  | BCEWithLogitsLoss | 0      | train
----------------------------------------------------
160 K     Trainable params
0         Non-trainable params
160 K     Total params
0.641     Total estimated model params size (MB)
```

```
6         Modules in train mode
0         Modules in eval mode

{"model_id":"0eaa8801c5384ee4b65677e1d9cc4481","version_major":2,"vers
ion_minor":0}

/usr/local/lib/python3.10/dist-packages/torch/utils/data/
dataloader.py:617: UserWarning: This DataLoader will create 6 worker
processes in total. Our suggested max number of worker in current
system is 2, which is smaller than what this DataLoader is going to
create. Please be aware that excessive worker creation might get
DataLoader running slow or even freeze, lower the worker number to
avoid potential slowness/freeze if necessary.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/pytorch_lightning/loops/fit_lo
op.py:298: The number of training batches (45) is smaller than the
logging interval Trainer(log_every_n_steps=50). Set a lower value for
log_every_n_steps if you want to see logs for the training epoch.

{"model_id":"37f4b9a705ce4381bf11b0015023b86a","version_major":2,"vers
ion_minor":0}

{"model_id":"d6d0270614fb4e44b805194db55137c5","version_major":2,"vers
ion_minor":0}

{"model_id":"b17c233863264f40ba032121c5e2a9b7","version_major":2,"vers
ion_minor":0}

{"model_id":"833c8bf9c5744dfabe48a1d525f193d4","version_major":2,"vers
ion_minor":0}

{"model_id":"f11aa95364c4448d8eaba04a402be1da","version_major":2,"vers
ion_minor":0}

{"model_id":"94ece0c4087842af98c550a42841e844","version_major":2,"vers
ion_minor":0}

{"model_id":"b140e44505de4b46a3f672c87e931394","version_major":2,"vers
ion_minor":0}

{"model_id":"cae569af6a454d4d97dc1ab8a500c6ae","version_major":2,"vers
ion_minor":0}

{"model_id":"49135355ceef44478f594a09d65a9a83","version_major":2,"vers
ion_minor":0}

{"model_id":"cd76286d2d5a41ab9064adee577fe21d","version_major":2,"vers
ion_minor":0}

{"model_id":"8c0496a88fbe4301a5f108fa4849589d","version_major":2,"vers
ion_minor":0}
```

```
{"model_id":"ff5c243ff9614f8f96fcf86ec5a58fb1","version_major":2,"version_minor":0}

{"model_id":"cf3d6c7d380e460bb5838f327d8a9491","version_major":2,"version_minor":0}

{"model_id":"2f384521222c4bb58c87544aead2f551","version_major":2,"version_minor":0}

{"model_id":"29480cfc1b4a4eab96004a07a70422a8","version_major":2,"version_minor":0}

{"model_id":"5cf9f0516d5547f9baf9670ba14d8a77","version_major":2,"version_minor":0}

{"model_id":"72c4362134534367b19184ace905bd24","version_major":2,"version_minor":0}

{"model_id":"8d2c551f3afb4682941df44f6f91ec67","version_major":2,"version_minor":0}

{"model_id":"59e0b52a358d4fefa7c61bae5f72d04f","version_major":2,"version_minor":0}

{"model_id":"a93a60406f58459b9aa24542ef6da4a0","version_major":2,"version_minor":0}

{"model_id":"055657871676495eb7f4c7ffc0e9fdee","version_major":2,"version_minor":0}

{"model_id":"85254e4470bb417eb95f282da0cd381a","version_major":2,"version_minor":0}

{"model_id":"238cb3323fdd49919716320b0f8746f6","version_major":2,"version_minor":0}

{"model_id":"2de5528d9d4543439f324bf8e05552f3","version_major":2,"version_minor":0}

{"model_id":"3638c6d606324fe7a0e5fde10887301d","version_major":2,"version_minor":0}

{"model_id":"10fef171263648c6b26d2532fdfa0165","version_major":2,"version_minor":0}

{"model_id":"d5efd92c1f964506beec774420d50184","version_major":2,"version_minor":0}

{"model_id":"f95063ef795941e88ec85fa47ab50463","version_major":2,"version_minor":0}

{"model_id":"dd961a8c61b24f59813534776f9c2f7e","version_major":2,"version_minor":0}
```

```
{"model_id":"8c4958621c1841658b65de018449bca7","version_major":2,"version_minor":0}
```

```
{"model_id":"3ca4701ba6c944e78bd59c2d2a6e75e8","version_major":2,"version_minor":0}
```

```
INFO:pytorch_lightning.utilities.rank_zero:`Trainer.fit` stopped: `max_epochs=30` reached.
```

```
{"model_id":"90d0f1d20d784f80804db1185a9a7dee","version_major":2,"version_minor":0}
```

```
┏━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━━━━┓
┃        Test metric        ┃       DataLoader 0        ┃
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━━━━┩
│       test_accuracy       │     0.853600025177002     │
│         test_loss         │    1.1887187957763672     │
└───────────────────────────┴───────────────────────────┘
```

```
[{'test_loss': 1.1887187957763672, 'test_accuracy': 0.853600025177002}]
```

```python
#from torch.nn import nn
# Data Loading from chunk above and extracting training and test set in
# sequential format
max_num_workers = 10
(imdb_seq_train, imdb_seq_test) = load_sequential(root='data/IMDB')
(imdb_train, imdb_test) = load_tensor(root='data/IMDB')

# Define a new class that inherits from the base class for all NN modules in PyTorch (nn.Module)
class IMDBModel(nn.Module):
    # Initialize the model's parameters as specified in the question with dropout rate for regularization
    def __init__(self, input_size, hidden_units = 16, dropout_rate = 0.0):
        super(IMDBModel, self).__init__()

        # Define layers with configurable hidden units and optional dropout
        self.network = nn.Sequential(
            nn.Linear(input_size, hidden_units), # A fully connected layer that maps the input features to the hidden units
            nn.ReLU(),  # ReLU activation function for non-linearity
            nn.Dropout(p = dropout_rate),  # Dropout layer to prevent overfitting
            # Above sequence is repeated with another fully connected layer, ReLU and dropout
            nn.Linear(hidden_units, hidden_units),
            nn.ReLU(),
            nn.Dropout(p = dropout_rate),
            # A fully connected layer that maps the hidden units to a
single output unit
            nn.Linear(hidden_units, 1))

    # This method defines the forward pass of the model with input
tensor x,
    # passes the input through the defined network layers above and
flattens the
    # output tensor to a 1D tensor.
    def forward(self, x):
        return torch.flatten(self.network(x))

# Define a function to train and evaluate model to do with and without
30 %
# dropout regularization.
def train_and_evaluate_model(hidden_units, dropout_rate):
    # Prepare DataModule to load data and to preprocess
    imdb_dm = SimpleDataModule(imdb_train, imdb_test, validation =
2000,
                               num_workers = min(6, max_num_workers),
                               batch_size = 512)
    # Create model using IMDB Model created above using number of
features in the test dataset as the input size
    imdb_model = IMDBModel(input_size = imdb_test.tensors[0].size()
[1],
                           hidden_units = hidden_units,
                           dropout_rate = dropout_rate)
    # Optimize using RMSProp and set learning rate to 0.01
    imdb_optimizer = RMSprop(imdb_model.parameters(), lr = 0.01)
    # Lightning Module for binary classifiaction using PyTorch
Lightning
    imdb_module = SimpleModule.binary_classification(imdb_model,
optimizer = imdb_optimizer)
    # Logger for saving training logs
    imdb_logger = CSVLogger('logs', name =
f'IMDB_units_{hidden_units}_dropout_{dropout_rate}')
    # Trainer handles the training loop to set the epochs
    imdb_trainer = Trainer(deterministic = True, max_epochs = 30,
                           logger = imdb_logger, callbacks =
[ErrorTracker()])
    # Train and Test the model using above functions
    imdb_trainer.fit(imdb_module, datamodule = imdb_dm)
    test_res = imdb_trainer.test(imdb_module, datamodule = imdb_dm)

    # Return results of model evaluation
    return test_res

# Experiment configurations categories to specify the number of hidden
units and dropout rate as specified in the question
```

```
exper_cat = [{'hidden_units': 16, 'dropout_rate': 0.0},   # Baseline
    {'hidden_units': 32, 'dropout_rate': 0.0},   # Increased hidden
units, no dropout
    {'hidden_units': 64, 'dropout_rate': 0.0},   # Increased hidden
units, no dropout
    {'hidden_units': 16, 'dropout_rate': 0.3},   # Baseline with
dropout
    {'hidden_units': 32, 'dropout_rate': 0.3},   # Increased hidden
units with 30 % dropout
    {'hidden_units': 64, 'dropout_rate': 0.3}    # Increased hidden
units with 30 % dropout
]

# Run experiments over each category above
res = {}
for exp in exper_cat:
    print(f"Running experiment: {exp}")
    res[f"{exp['hidden_units']} units, {exp['dropout_rate']} dropout"]
= train_and_evaluate_model(
        exp['hidden_units'],
        exp['dropout_rate']
    )

# Print res (Note: actual print formatting might need adjustment)
for config, result in res.items():
    print(f"\nConfiguration: {config}")
    print(result)
```

```
/usr/local/lib/python3.10/dist-packages/ISLP/torch/imdb.py:131:
FutureWarning: You are using `torch.load` with `weights_only=False`
(the current default value), which uses the default pickle module
implicitly. It is possible to construct malicious pickle data which
will execute arbitrary code during unpickling (See
https://github.com/pytorch/pytorch/blob/main/SECURITY.md#untrusted-
models for more details). In a future release, the default value for
`weights_only` will be flipped to `True`. This limits the functions
that could be executed during unpickling. Arbitrary objects will no
longer be allowed to be loaded via this mode unless they are
explicitly allowlisted by the user via
`torch.serialization.add_safe_globals`. We recommend you start setting
`weights_only=True` for any use case where you don't have full control
of the loaded file. Please open an issue on GitHub for any issues
related to this experimental feature.
  S_test) = [torch.load(_get_imdb(f'IMDB_{r}', root))
/usr/local/lib/python3.10/dist-packages/ISLP/torch/imdb.py:113:
FutureWarning: You are using `torch.load` with `weights_only=False`
(the current default value), which uses the default pickle module
implicitly. It is possible to construct malicious pickle data which
will execute arbitrary code during unpickling (See
https://github.com/pytorch/pytorch/blob/main/SECURITY.md#untrusted-
```

```
models for more details). In a future release, the default value for
`weights_only` will be flipped to `True`. This limits the functions
that could be executed during unpickling. Arbitrary objects will no
longer be allowed to be loaded via this mode unless they are
explicitly allowlisted by the user via
`torch.serialization.add_safe_globals`. We recommend you start setting
`weights_only=True` for any use case where you don't have full control
of the loaded file. Please open an issue on GitHub for any issues
related to this experimental feature.
  X_test, X_train = [torch.load(_get_imdb(f'IMDB_{r}', root))
INFO:pytorch_lightning.utilities.rank_zero:GPU available: False, used:
False
INFO:pytorch_lightning.utilities.rank_zero:TPU available: False,
using: 0 TPU cores
INFO:pytorch_lightning.utilities.rank_zero:HPU available: False,
using: 0 HPUs
INFO:pytorch_lightning.callbacks.model_summary:
  | Name  | Type              | Params | Mode
----------------------------------------------------
0 | model | IMDBModel         | 160 K  | train
1 | loss  | BCEWithLogitsLoss | 0      | train
----------------------------------------------------
160 K     Trainable params
0         Non-trainable params
160 K     Total params
0.641     Total estimated model params size (MB)
10        Modules in train mode
0         Modules in eval mode

Running experiment: {'hidden_units': 16, 'dropout_rate': 0.0}

{"model_id":"661e7b96108c4e0eba5daa6d2e1300fd","version_major":2,"vers
ion_minor":0}

/usr/local/lib/python3.10/dist-packages/torch/utils/data/
dataloader.py:617: UserWarning: This DataLoader will create 6 worker
processes in total. Our suggested max number of worker in current
system is 2, which is smaller than what this DataLoader is going to
create. Please be aware that excessive worker creation might get
DataLoader running slow or even freeze, lower the worker number to
avoid potential slowness/freeze if necessary.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/pytorch_lightning/loops/fit_lo
op.py:298: The number of training batches (45) is smaller than the
logging interval Trainer(log_every_n_steps=50). Set a lower value for
log_every_n_steps if you want to see logs for the training epoch.

{"model_id":"84a73b3b54be43ef98b0505715aa4a4b","version_major":2,"vers
ion_minor":0}
```

```
{"model_id":"ff90b07e16c14a9c83bc6c27a6ffcc3a","version_major":2,"version_minor":0}

{"model_id":"5a0e7e9f4f5a419490369db09ed75f74","version_major":2,"version_minor":0}

{"model_id":"d5e088c013a840c1a1755ce05309d505","version_major":2,"version_minor":0}

{"model_id":"4c3345a00f964c4ba2e9879cd55899ee","version_major":2,"version_minor":0}

{"model_id":"b0c3ef02cf574cc095738467b49291ed","version_major":2,"version_minor":0}

{"model_id":"d95ae31ced6d4c1986c61501238eb45c","version_major":2,"version_minor":0}

{"model_id":"b54f41e9996a4f508c6eef1b32372017","version_major":2,"version_minor":0}

{"model_id":"b6d26a0554884b65bed0c5aaa2363d33","version_major":2,"version_minor":0}

{"model_id":"39b25744eef243ee95c37cb36ec23037","version_major":2,"version_minor":0}

{"model_id":"b30e19e78dc2420aa4d12be1a4f88611","version_major":2,"version_minor":0}

{"model_id":"cdaa3332eb6e40fb89b9eea69c5334cd","version_major":2,"version_minor":0}

{"model_id":"20fc0aec3ad647389c265e5b252f62b8","version_major":2,"version_minor":0}

{"model_id":"35b0afe076d140fea31bdebd855589ed","version_major":2,"version_minor":0}

{"model_id":"acc3a611bf88497b8851dc389f515cf5","version_major":2,"version_minor":0}

{"model_id":"f52ca07d679f4d74ac7736db94e1bb99","version_major":2,"version_minor":0}

{"model_id":"f219dd5f27a945f885fd7c5406e85c97","version_major":2,"version_minor":0}

{"model_id":"4b766343a28c47ddacf2f1da71d8004a","version_major":2,"version_minor":0}

{"model_id":"385c08fa010f4637aabc0c0f59f00802","version_major":2,"version_minor":0}
```

```
{"model_id":"292f7aef07054ed0884448817cf7aee2","version_major":2,"version_minor":0}

{"model_id":"609b332fff8d4085b1ba2b1f26d14029","version_major":2,"version_minor":0}

{"model_id":"e63364da75694a70aca2a6cd770a5eb0","version_major":2,"version_minor":0}

{"model_id":"b42ccc02383143758909ba45e02423e0","version_major":2,"version_minor":0}

{"model_id":"7c78b39ca6fe45dd8046b0dc71957f44","version_major":2,"version_minor":0}

{"model_id":"1af03cb79a0748a2a34c3d8875abd688","version_major":2,"version_minor":0}

{"model_id":"db00805c163a4d51a54f12333deaf780","version_major":2,"version_minor":0}

{"model_id":"b46e1748f5ae4d98971a3122d5bc712a","version_major":2,"version_minor":0}

{"model_id":"546de2ee936f46868b5258e60c826c38","version_major":2,"version_minor":0}

{"model_id":"caa87efbc8764d99a55f124c2b0507f4","version_major":2,"version_minor":0}

{"model_id":"b3e0dd63dbc94308a9a9c6656d7e99d0","version_major":2,"version_minor":0}

{"model_id":"7e87db2126434968a3a5949a37798a17","version_major":2,"version_minor":0}

INFO:pytorch_lightning.utilities.rank_zero:`Trainer.fit` stopped:
`max_epochs=30` reached.

{"model_id":"bd80cc85fdc145ed9f243caa3ad85d7e","version_major":2,"version_minor":0}

┏━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━━━━┓
┃        Test metric        ┃       DataLoader 0        ┃
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━━━━┩
│       test_accuracy       │    0.8548799753189087     │
│         test_loss         │     1.325154423713684     │
└───────────────────────────┴───────────────────────────┘

Running experiment: {'hidden_units': 32, 'dropout_rate': 0.0}

INFO:pytorch_lightning.utilities.rank_zero:GPU available: False, used:
False
```

```
INFO:pytorch_lightning.utilities.rank_zero:TPU available: False, using: 0 TPU cores
INFO:pytorch_lightning.utilities.rank_zero:HPU available: False, using: 0 HPUs
INFO:pytorch_lightning.callbacks.model_summary:
  | Name  | Type              | Params | Mode 
----------------------------------------------------
0 | model | IMDBModel         | 321 K  | train
1 | loss  | BCEWithLogitsLoss | 0      | train
----------------------------------------------------
321 K     Trainable params
0         Non-trainable params
321 K     Total params
1.285     Total estimated model params size (MB)
10        Modules in train mode
0         Modules in eval mode
```

```
{"model_id":"4f8993d573464f47815e3c18df9e0530","version_major":2,"version_minor":0}
```

```
{"model_id":"f8adc9bc8a9743e0b5f0c62fa55b22c7","version_major":2,"version_minor":0}
```

```
{"model_id":"4db497546ca242b2ba0883805a4f3348","version_major":2,"version_minor":0}
```

```
{"model_id":"c2167f2f2d944175a9f45e160115e53d","version_major":2,"version_minor":0}
```

```
{"model_id":"9d20d24af56f46578bd6b2bd83dab7bd","version_major":2,"version_minor":0}
```

```
{"model_id":"9a2bac09208747dc8c2ac88ac70de13c","version_major":2,"version_minor":0}
```

```
{"model_id":"41f9cb6fd59145b98215ff3df6cd1e20","version_major":2,"version_minor":0}
```

```
{"model_id":"229af18222114b7ebea609647b259ee0","version_major":2,"version_minor":0}
```

```
{"model_id":"d682399c74014955bb6fa3e6629984f1","version_major":2,"version_minor":0}
```

```
{"model_id":"f99e6f51f6e64ca39d69a5808a492b8d","version_major":2,"version_minor":0}
```

```
{"model_id":"36430914e4d9470b931b8d11be027917","version_major":2,"version_minor":0}
```

```
{"model_id":"513fc8f2de5f42fdb54165024efd1375","version_major":2,"version_minor":0}
```

```
{"model_id":"71be22ae606e481f9fded6d0e008d9bc","version_major":2,"version_minor":0}

{"model_id":"41a3c2e3b3e34390b62f14e541a140dd","version_major":2,"version_minor":0}

{"model_id":"a1aabb35dc7d45629a6a6456e12a1ce2","version_major":2,"version_minor":0}

{"model_id":"7e31ead53aae4db9bfb8f41fa2b27427","version_major":2,"version_minor":0}

{"model_id":"29aa45edd88f48bab8f46b7617115518","version_major":2,"version_minor":0}

{"model_id":"7de78e0681d44c0881946a310c010fd6","version_major":2,"version_minor":0}

{"model_id":"e09054a25a98453bbc94feb637b4801c","version_major":2,"version_minor":0}

{"model_id":"1ea70d3e082441fcbb35cb5878d11ac7","version_major":2,"version_minor":0}

{"model_id":"64374b2f33b947c1945d6383398d0da0","version_major":2,"version_minor":0}

{"model_id":"c7e8ba2119364a26b336b0791c9d173e","version_major":2,"version_minor":0}

{"model_id":"4f00693fdf2a4d4b9b3b0edafc6a573c","version_major":2,"version_minor":0}

{"model_id":"1456f962cf5c4882bf24f6fb2248ed5e","version_major":2,"version_minor":0}

{"model_id":"6b12d46b771f47448e6c58d030c14b4c","version_major":2,"version_minor":0}

{"model_id":"4ba41c3ce5074e18ab3871c4cada4602","version_major":2,"version_minor":0}

{"model_id":"1ac4f2a96cbd4cabb1859eabda974f96","version_major":2,"version_minor":0}

{"model_id":"feea97f2e9994d458e5c2075e7295d2d","version_major":2,"version_minor":0}

{"model_id":"983ad0759ea64c33bb75703527b28468","version_major":2,"version_minor":0}

{"model_id":"d440ea39646b438287a5833c874db9b4","version_major":2,"version_minor":0}
```

```
{"model_id":"0ccfb4f0de104e6ea059e508f8d88222","version_major":2,"version_minor":0}

{"model_id":"5b7dc62d612144dba18a271424afedd9","version_major":2,"version_minor":0}

INFO:pytorch_lightning.utilities.rank_zero:`Trainer.fit` stopped: `max_epochs=30` reached.

{"model_id":"5a4574505f9b490cadc2e5b1dc26181e","version_major":2,"version_minor":0}

┏━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━━━━┓
┃        Test metric        ┃       DataLoader 0        ┃
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━━━━┩
│       test_accuracy       │    0.8522800207138062     │
│         test_loss         │    1.5540151596069336     │
└───────────────────────────┴───────────────────────────┘

INFO:pytorch_lightning.utilities.rank_zero:GPU available: False, used: False
INFO:pytorch_lightning.utilities.rank_zero:TPU available: False, using: 0 TPU cores
INFO:pytorch_lightning.utilities.rank_zero:HPU available: False, using: 0 HPUs
INFO:pytorch_lightning.callbacks.model_summary:
  | Name  | Type              | Params | Mode 
----------------------------------------------------
0 | model | IMDBModel         | 644 K  | train
1 | loss  | BCEWithLogitsLoss | 0      | train
----------------------------------------------------
644 K     Trainable params
0         Non-trainable params
644 K     Total params
2.578     Total estimated model params size (MB)
10        Modules in train mode
0         Modules in eval mode

Running experiment: {'hidden_units': 64, 'dropout_rate': 0.0}

{"model_id":"9a610d56a7394726b87b41854b7860d8","version_major":2,"version_minor":0}

{"model_id":"ec478e93146b4076aa566a3cb095f5d7","version_major":2,"version_minor":0}

{"model_id":"f6cb1f4d944b41a7aedc01d62ceb9cb0","version_major":2,"version_minor":0}

{"model_id":"9cea51972ebf42cb905a2c0d7e064216","version_major":2,"version_minor":0}
```

```
{"model_id":"a7ca1f68f2a74d8f9543b0e634ddd392","version_major":2,"version_minor":0}

{"model_id":"49557ca5dab5415e9028678b4c8d3812","version_major":2,"version_minor":0}

{"model_id":"bfda613f3e014f86b1988d0a469a5147","version_major":2,"version_minor":0}

{"model_id":"c5da2854c5a14c6db12af0bf419533cb","version_major":2,"version_minor":0}

{"model_id":"ac357a147a774cc2a5026dc4ef9a8c21","version_major":2,"version_minor":0}

{"model_id":"ddc5c34f47de4acc9820f173feeb4a61","version_major":2,"version_minor":0}

{"model_id":"6d923a9b3ba8416d98faf40c736452b1","version_major":2,"version_minor":0}

{"model_id":"b4bedc6d6dea41a8b1f6f59ecc384040","version_major":2,"version_minor":0}

{"model_id":"cb2abcd8ccae42bdb8efed1748405635","version_major":2,"version_minor":0}

{"model_id":"708e003cf5904fd58b31007eb02e2b7b","version_major":2,"version_minor":0}

{"model_id":"5d5c5932ab354f4a9a643517564f02a4","version_major":2,"version_minor":0}

{"model_id":"810ecbd2c62447d889e02259f18868dc","version_major":2,"version_minor":0}

{"model_id":"ef555606b95c4dc8afe6218cdfe6966d","version_major":2,"version_minor":0}

{"model_id":"6d5380ef3bf44e80893d0f261cfa9afb","version_major":2,"version_minor":0}

{"model_id":"24d36076a9c9480892cc16477add0213","version_major":2,"version_minor":0}

{"model_id":"6066e5bb09624c988aa5d5a662704842","version_major":2,"version_minor":0}

{"model_id":"12a0ca47249f416784922085a11ece4c","version_major":2,"version_minor":0}

{"model_id":"f8e630e05416435586d65d1aad924a52","version_major":2,"version_minor":0}
```

```
{"model_id":"9fb6a8172a3c4b3dab87e42355777eeb","version_major":2,"vers
ion_minor":0}

{"model_id":"d32371e89d0a4a14b8e771f3dd693219","version_major":2,"vers
ion_minor":0}

{"model_id":"906b0e585f354c41a2207d5c5b898079","version_major":2,"vers
ion_minor":0}

{"model_id":"c4e179cf495242c69196566d30ad151f","version_major":2,"vers
ion_minor":0}

{"model_id":"52c47713b1f942bc91905fe8934cfdde","version_major":2,"vers
ion_minor":0}

{"model_id":"c3ee4ac027d04800ad602e0ca731e033","version_major":2,"vers
ion_minor":0}

{"model_id":"3f91f24060c84b669b3cc519e7f915ed","version_major":2,"vers
ion_minor":0}

{"model_id":"924cedf6e11b4567a36cf30787b567f5","version_major":2,"vers
ion_minor":0}

{"model_id":"39389047b1b842bb8e930d0b38a41992","version_major":2,"vers
ion_minor":0}

{"model_id":"01f41b256b884903bced320efe6ad200","version_major":2,"vers
ion_minor":0}

INFO:pytorch_lightning.utilities.rank_zero:`Trainer.fit` stopped:
`max_epochs=30` reached.

{"model_id":"1bdee41edb9948f59ac4c72f0ea02a2e","version_major":2,"vers
ion_minor":0}
```

Test metric	DataLoader 0
test_accuracy test_loss	0.8658000230789185 1.1865366697311401

```
INFO:pytorch_lightning.utilities.rank_zero:GPU available: False, used:
False
INFO:pytorch_lightning.utilities.rank_zero:TPU available: False,
using: 0 TPU cores
INFO:pytorch_lightning.utilities.rank_zero:HPU available: False,
using: 0 HPUs
INFO:pytorch_lightning.callbacks.model_summary:
  | Name  | Type              | Params | Mode
----------------------------------------------------
```

```
0 | model | IMDBModel         | 160 K  | train
1 | loss  | BCEWithLogitsLoss | 0      | train
----------------------------------------------------
160 K     Trainable params
0         Non-trainable params
160 K     Total params
0.641     Total estimated model params size (MB)
10        Modules in train mode
0         Modules in eval mode

Running experiment: {'hidden_units': 16, 'dropout_rate': 0.3}

{"model_id":"24bda01d4ab04bcb9a2b6a2e19b8747a","version_major":2,"version_minor":0}

{"model_id":"b13c0fb3983d48fea69d29695560d279","version_major":2,"version_minor":0}

{"model_id":"a312e0fd00ec4a54be2269692365570b","version_major":2,"version_minor":0}

{"model_id":"6bcb5ede3a594c7389a8039c39e70363","version_major":2,"version_minor":0}

{"model_id":"a5843d867c24427d9cad9fc0b93fb535","version_major":2,"version_minor":0}

{"model_id":"9430f88d156d41fa96415774d8aeca45","version_major":2,"version_minor":0}

{"model_id":"b29b776f12854a0993dd16191f105ca6","version_major":2,"version_minor":0}

{"model_id":"079d745ac52c46b6a370d311b3a5a317","version_major":2,"version_minor":0}

{"model_id":"1db320b8b94a49ab88aa7633a531f296","version_major":2,"version_minor":0}

{"model_id":"b3625f9e778a4e038465eb15560d83ad","version_major":2,"version_minor":0}

{"model_id":"a4af9975f3aa422ab0c176037b406853","version_major":2,"version_minor":0}

{"model_id":"9efb2173e7b240118c8d696114cfb782","version_major":2,"version_minor":0}

{"model_id":"b45c50595d614081a832ebd698e4501c","version_major":2,"version_minor":0}

{"model_id":"95ccb4cb09b14d999c3dc588937e2cae","version_major":2,"version_minor":0}
```

```
{"model_id":"d76fae8747814715ae63d9b3ff130693","version_major":2,"version_minor":0}

{"model_id":"7617ed30b8714a00859af50e832e6e68","version_major":2,"version_minor":0}

{"model_id":"d575d449ed5b45178474eeb5a8a87080","version_major":2,"version_minor":0}

{"model_id":"d9c18d79c97844c4b41fe8e1aca9bc2d","version_major":2,"version_minor":0}

{"model_id":"e02904977ca84fb998216b1b7a3135a7","version_major":2,"version_minor":0}

{"model_id":"37ce37b46f1e4fb186b55a1fd588aabc","version_major":2,"version_minor":0}

{"model_id":"fa2dec1a593e4b4c8fa6a3b467f9e839","version_major":2,"version_minor":0}

{"model_id":"6c434ea669984c53b7e814f3653a1fdf","version_major":2,"version_minor":0}

{"model_id":"581a13996d5245f2a924d3b836a0b7e2","version_major":2,"version_minor":0}

{"model_id":"335aa7c177bc4ea0b63681f138128df4","version_major":2,"version_minor":0}

{"model_id":"e1744316dc9047ddb2eea1b1fef91b45","version_major":2,"version_minor":0}

{"model_id":"ab3ddba4211c4d28bdc93f7549684c5c","version_major":2,"version_minor":0}

{"model_id":"cda854cce9fb43ffb438404d478f548c","version_major":2,"version_minor":0}

{"model_id":"90fe1973509d456db653fc7255639464","version_major":2,"version_minor":0}

{"model_id":"92f5da876b1747d9865929f113b533d8","version_major":2,"version_minor":0}

{"model_id":"4eb5f0a0c676451986b532ea7ea78533","version_major":2,"version_minor":0}

{"model_id":"43290918571b47a5aaef2773b6ee7f82","version_major":2,"version_minor":0}

{"model_id":"f7f61a1cd2924277b6af6964fd13ba0a","version_major":2,"version_minor":0}
```

```
INFO:pytorch_lightning.utilities.rank_zero:`Trainer.fit` stopped:
`max_epochs=30` reached.
```

{"model_id":"ddaf57c3b10a4061a140672c81acb5b1","version_major":2,"vers
ion_minor":0}

```
┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━━━━━┓
┃        Test metric         ┃        DataLoader 0        ┃
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━━━━━┩
│       test_accuracy        │     0.8689600229263306     │
│         test_loss          │     1.3017518520355225     │
└────────────────────────────┴────────────────────────────┘
```

```
INFO:pytorch_lightning.utilities.rank_zero:GPU available: False, used:
False
INFO:pytorch_lightning.utilities.rank_zero:TPU available: False,
using: 0 TPU cores
INFO:pytorch_lightning.utilities.rank_zero:HPU available: False,
using: 0 HPUs
INFO:pytorch_lightning.callbacks.model_summary:
  | Name  | Type              | Params | Mode
----------------------------------------------------
0 | model | IMDBModel         | 321 K  | train
1 | loss  | BCEWithLogitsLoss | 0      | train
----------------------------------------------------
321 K     Trainable params
0         Non-trainable params
321 K     Total params
1.285     Total estimated model params size (MB)
10        Modules in train mode
0         Modules in eval mode
```

```
Running experiment: {'hidden_units': 32, 'dropout_rate': 0.3}
```

{"model_id":"bda7a78b77cd46ed87adc71d1b0cc686","version_major":2,"vers
ion_minor":0}

{"model_id":"9054742b09984822958827a9e75549ce","version_major":2,"vers
ion_minor":0}

{"model_id":"2881a59a24c34aefaf2ad7476e78940b","version_major":2,"vers
ion_minor":0}

{"model_id":"fd3bf5c9fb2e406680d9626033050096","version_major":2,"vers
ion_minor":0}

{"model_id":"a1e022a0a0274e5885874a19513b364e","version_major":2,"vers
ion_minor":0}

{"model_id":"bf6e3d030c8f4c9b87c127a1853f1bb2","version_major":2,"vers
ion_minor":0}

```
{"model_id":"3bbd70cb64b6497fa9cd3edfeefd52a3","version_major":2,"version_minor":0}

{"model_id":"159406c42c574189a734c4fce8d7a26f","version_major":2,"version_minor":0}

{"model_id":"ce8077e5ea614c35b5a74fc832b7dbb8","version_major":2,"version_minor":0}

{"model_id":"abc735cb56f64b4dbf4dcbfcc230088b","version_major":2,"version_minor":0}

{"model_id":"14fd5b657f77456db2d0a1496e60d3fd","version_major":2,"version_minor":0}

{"model_id":"cce182053806474c99a56d664a2da903","version_major":2,"version_minor":0}

{"model_id":"6f687a8690cc42158c03c62ca37f4002","version_major":2,"version_minor":0}

{"model_id":"00136c6efb9744f487f9c90de9ec45ca","version_major":2,"version_minor":0}

{"model_id":"b85b464dcf67403cbe33d2e59452d421","version_major":2,"version_minor":0}

{"model_id":"f5659c072f404b9790a8bbe2ff440447","version_major":2,"version_minor":0}

{"model_id":"1cf526c419464fe9bf41fe7179b19855","version_major":2,"version_minor":0}

{"model_id":"d579843d68124443a54c592cafd405f9","version_major":2,"version_minor":0}

{"model_id":"9a9eaeb9b93b422ebee1e2b4dc84cefd","version_major":2,"version_minor":0}

{"model_id":"e768522490204636957ee5cf8d8a7e45","version_major":2,"version_minor":0}

{"model_id":"feccd826c4834e149a1d0099dfeb741f","version_major":2,"version_minor":0}

{"model_id":"e358fbf57e3f474db06f3ade32dccd1e","version_major":2,"version_minor":0}

{"model_id":"2ac46c31320f48149840937ed5b80199","version_major":2,"version_minor":0}

{"model_id":"a85178b5b5734ac79c79b6e0285e6bb3","version_major":2,"version_minor":0}
```

```
{"model_id":"4a0c73a99e5c4bc88ecc6ce0c976864b","version_major":2,"version_minor":0}

{"model_id":"f32870b6e5c74737a6cc9ab3bc440fb0","version_major":2,"version_minor":0}

{"model_id":"5379a42241d1435b8fbf60ee196025a8","version_major":2,"version_minor":0}

{"model_id":"a0471e60d18043cb8f138c4691cbf8f4","version_major":2,"version_minor":0}

{"model_id":"83e74bf797b24ed9802104046ce20dae","version_major":2,"version_minor":0}

{"model_id":"b67e67b0f79947baa010d3938781260b","version_major":2,"version_minor":0}

{"model_id":"ef66626c1e8647d387cb375054fdde2b","version_major":2,"version_minor":0}

{"model_id":"1f2ca3aaa99f4f889b9160c904d19e30","version_major":2,"version_minor":0}

INFO:pytorch_lightning.utilities.rank_zero:`Trainer.fit` stopped:
`max_epochs=30` reached.

{"model_id":"927a11e155984eb89fce195e29e43f18","version_major":2,"version_minor":0}

┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━━━━━┓
┃        Test metric         ┃        DataLoader 0        ┃
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━━━━━┩
│       test_accuracy        │     0.8685600161552429     │
│         test_loss          │     0.8540134429931641     │
└────────────────────────────┴────────────────────────────┘

INFO:pytorch_lightning.utilities.rank_zero:GPU available: False, used:
False
INFO:pytorch_lightning.utilities.rank_zero:TPU available: False,
using: 0 TPU cores
INFO:pytorch_lightning.utilities.rank_zero:HPU available: False,
using: 0 HPUs
INFO:pytorch_lightning.callbacks.model_summary:
  | Name  | Type              | Params | Mode
---------------------------------------------------
0 | model | IMDBModel         | 644 K  | train
1 | loss  | BCEWithLogitsLoss | 0      | train
---------------------------------------------------
644 K     Trainable params
0         Non-trainable params
644 K     Total params
```

```
2.578     Total estimated model params size (MB)
10        Modules in train mode
0         Modules in eval mode

Running experiment: {'hidden_units': 64, 'dropout_rate': 0.3}

{"model_id":"c45714a10156495faadf65ce31e5797b","version_major":2,"vers
ion_minor":0}

{"model_id":"465dcf17b14949389f417833be5c1e35","version_major":2,"vers
ion_minor":0}

{"model_id":"ef98c73d57814107b9f7a07f1ac58ae2","version_major":2,"vers
ion_minor":0}

{"model_id":"6ad02fcde76a4ca7a9bf47269f50f706","version_major":2,"vers
ion_minor":0}

{"model_id":"849d9b5a22064bac826e895337f985db","version_major":2,"vers
ion_minor":0}

{"model_id":"961c4f9c057648ab859ae8d2996ba2e4","version_major":2,"vers
ion_minor":0}

{"model_id":"66166258a83e4af98a6c07d6f2c39b04","version_major":2,"vers
ion_minor":0}

{"model_id":"e347d496b8364d4a9f0c3d706c900143","version_major":2,"vers
ion_minor":0}

{"model_id":"0823ed315dcd4a6ea1d08a4d6ccc885b","version_major":2,"vers
ion_minor":0}

{"model_id":"8abcfcd9b3ed40578e61a02e30e996cf","version_major":2,"vers
ion_minor":0}

{"model_id":"764f48f8aeb14a54b7b071f859b30d45","version_major":2,"vers
ion_minor":0}

{"model_id":"45494d23b97546ac9f622f835b503a00","version_major":2,"vers
ion_minor":0}

{"model_id":"7f5d5557d87a4147ba8c1ada7aa0d4d5","version_major":2,"vers
ion_minor":0}

{"model_id":"087d4a03bf06475d8b20adac7915e8fb","version_major":2,"vers
ion_minor":0}

{"model_id":"90ea74a6bd984b3e958ce63cbad130e1","version_major":2,"vers
ion_minor":0}

{"model_id":"395300a25c664445835b0e5cdfeb0a69","version_major":2,"vers
ion_minor":0}
```

```
{"model_id":"b982fcbf0d444739ae9bc7b4aaae138b","version_major":2,"vers
ion_minor":0}

{"model_id":"c1ded6aa51ab4936b8106e3209fbc354","version_major":2,"vers
ion_minor":0}

{"model_id":"5a05baccdd1e4a4482b03f6b67b28f81","version_major":2,"vers
ion_minor":0}

{"model_id":"d47e12db5b5f470d84f52b36b7e68713","version_major":2,"vers
ion_minor":0}

{"model_id":"5409e5239d3e4d04b00c0c9e9e5d61cd","version_major":2,"vers
ion_minor":0}

{"model_id":"2afb3763b94940388c59d57975ab2503","version_major":2,"vers
ion_minor":0}

{"model_id":"795f01bb8f004eaeb1895681a931df61","version_major":2,"vers
ion_minor":0}

{"model_id":"1a03595ff2584c7ea8d445274bbee9a8","version_major":2,"vers
ion_minor":0}

{"model_id":"0f40f944309548b1abf9b37c7422c7cc","version_major":2,"vers
ion_minor":0}

{"model_id":"2bff69a7ce9243b990850267ee84e7ea","version_major":2,"vers
ion_minor":0}

{"model_id":"da4bf41e25414ec2b0ff90b6940b4d2f","version_major":2,"vers
ion_minor":0}

{"model_id":"2ef5f9f4306a440d9466aea92afe8a2e","version_major":2,"vers
ion_minor":0}

{"model_id":"4d3e516e41d74811b1d29799433c1387","version_major":2,"vers
ion_minor":0}

{"model_id":"7bac574ec86e4901ae9bbeae9dab5de0","version_major":2,"vers
ion_minor":0}

{"model_id":"2b858ea605ef43ab98603fa05b14d56e","version_major":2,"vers
ion_minor":0}

{"model_id":"64a796f60b9647879f5af4e73658f77c","version_major":2,"vers
ion_minor":0}

INFO:pytorch_lightning.utilities.rank_zero:`Trainer.fit` stopped:
`max_epochs=30` reached.

{"model_id":"7804b416344f4436a7a97ac3f3137d17","version_major":2,"vers
ion_minor":0}
```

```
┏━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━━━━┓
┃        Test metric        ┃       DataLoader 0        ┃
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━━━━┩
│       test_accuracy       │     0.866919994354248     │
│         test_loss         │    0.8414745926856995     │
└───────────────────────────┴───────────────────────────┘


Configuration: 16 units, 0.0 dropout
[{'test_loss': 1.325154423713684, 'test_accuracy':
0.8548799753189087}]

Configuration: 32 units, 0.0 dropout
[{'test_loss': 1.5540151596069336, 'test_accuracy':
0.8522800207138062}]

Configuration: 64 units, 0.0 dropout
[{'test_loss': 1.1865366697311401, 'test_accuracy':
0.8658000230789185}]

Configuration: 16 units, 0.3 dropout
[{'test_loss': 1.3017518520355225, 'test_accuracy':
0.8689600229263306}]

Configuration: 32 units, 0.3 dropout
[{'test_loss': 0.8540134429931641, 'test_accuracy':
0.8685600161552429}]

Configuration: 64 units, 0.3 dropout
[{'test_loss': 0.8414745926856995, 'test_accuracy':
0.866919994354248}]
```