

[QA 요청] 실험 생성 코드 예시

- Example Code
 - Code Block
 - Parameters Block
 - Content
 - Input Example
 - Model Block
 - Content
 - Input Example
 - Data Block
 - Content
 - Input Example
 - Metric Block
 - Content
 - Input Example
-

Example Code

- 05_iris_tensorflow.ipynb
- 06_mnist_tensorflow.ipynb

Code Block

예시는 05_iris_tensorflow.ipynb 기준으로 작성하겠습니다.

아래는 코드 전문입니다.

```
import argparse
import os
import numpy as np

from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OneHotEncoder

import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.optimizers import Adam

def iris_model():
    model = Sequential()
    model.add(Dense(10, input_shape=(4,), activation='relu', name='fc1'))
    model.add(Dense(10, activation='relu', name='fc2'))
    model.add(Dense(3, activation='softmax', name='output'))

    return model

class MetricsPrint(tf.keras.callbacks.Callback):
    def on_epoch_end(self, epoch, logs=None):
```

```

"""
Simple function for printing the history so that Katib picks it up
"""

hist = self.model.history.history
history_keys = list(hist.keys())
print('Wnepoche {}'.format(epoch))
for cur_key in history_keys:
    print('{}={}'.format(cur_key, hist[cur_key][-1]))

def main():
    parser = argparse.ArgumentParser()
    parser.add_argument('--batch-size', type=int, default=5,
                        help='input batch size for training (default: 5)')
    parser.add_argument('--learning-rate', type=float, default=0.001,
                        help='learning rate (default: 0.001)')
    parser.add_argument('--epochs', type=int, default=100, metavar='N',
                        help='number of epochs to train (default: 100)')
    args = parser.parse_args(args=[])

    iris_data = load_iris() # load the iris dataset

    print('Example data: ')
    print(iris_data.data[:5])
    print('Example labels: ')
    print(iris_data.target[:5])

    x = iris_data.data
    y_ = iris_data.target.reshape(-1, 1) # Convert data to a single column

    # One Hot encode the class labels
    encoder = OneHotEncoder(sparse=False)
    y = encoder.fit_transform(y_)

    train_x, test_x, train_y, test_y = train_test_split(x, y, test_size=0.20)

    model = iris_model()

    # Adam optimizer with learning rate of 0.001
    optimizer = Adam(learning_rate=args.learning_rate)
    model.compile(optimizer, loss='categorical_crossentropy', metrics=['accuracy'])

    print('Neural Network Model Summary: ')
    print(model.summary())

    # Train the model
    history = model.fit(train_x, train_y, verbose=0, batch_size=args.batch_size, epochs=args.
epochs, callbacks=[MetricsPrint()])

if __name__ == "__main__":
    main()

```

Parameters Block

Content

main 블록 상의 parser.add_argument 항목 (원하는 항목 선택)

모든 항목 적용이 필수는 아닙니다.

선택한 항목이 실험 상의 하이퍼파라미터 항목으로 들어갑니다.

반드시 'add_argument(' 코드의 시작 괄호가 닫히도록 설정이 필요합니다.

Input Example

```
parser.add_argument('--batch-size', type=int, default=5,  
                    help='input batch size for training (default: 5)')  
parser.add_argument('--learning-rate', type=float, default=0.001,  
                    help='learning rate (default: 0.001)')
```

Model Block

Content

main block상단 모든 코드 (import 포함)

Input Example

```

import argparse
import os
import numpy as np

from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OneHotEncoder

import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.optimizers import Adam

def iris_model():
    model = Sequential()
    model.add(Dense(10, input_shape=(4,), activation='relu', name='fc1'))
    model.add(Dense(10, activation='relu', name='fc2'))
    model.add(Dense(3, activation='softmax', name='output'))

    return model

class MetricsPrint(tf.keras.callbacks.Callback):
    def on_epoch_end(self, epoch, logs=None):
        """
        Simple function for printing the history so that Katib picks it up
        """
        hist = self.model.history.history
        history_keys = list(hist.keys())
        print('Wn epoche {}'.format(epoch))
        for cur_key in history_keys:
            print('{}={}'.format(cur_key, hist[cur_key][-1]))

```

Data Block

Content

main block 하위 모든 코드

반드시 아래 if 라인이 들어가야 합니다.

```

if __name__ == "__main__":
    main()

```

Input Example

```

def main():
    parser = argparse.ArgumentParser()
    parser.add_argument('--batch-size', type=int, default=5,
                        help='input batch size for training (default: 5)')
    parser.add_argument('--learning-rate', type=float, default=0.001,
                        help='learning rate (default: 0.001)')
    parser.add_argument('--epochs', type=int, default=100, metavar='N',
                        help='number of epochs to train (default: 100)')
    args = parser.parse_args(args=[])

    iris_data = load_iris() # load the iris dataset

    print('Example data: ')
    print(iris_data.data[:5])
    print('Example labels: ')
    print(iris_data.target[:5])

    x = iris_data.data
    y_ = iris_data.target.reshape(-1, 1) # Convert data to a single column

    # One Hot encode the class labels
    encoder = OneHotEncoder(sparse=False)
    y = encoder.fit_transform(y_)

    train_x, test_x, train_y, test_y = train_test_split(x, y, test_size=0.20)

    model = iris_model()

    # Adam optimizer with learning rate of 0.001
    optimizer = Adam(learning_rate=args.learning_rate)
    model.compile(optimizer, loss='categorical_crossentropy', metrics=['accuracy'])

    print('Neural Network Model Summary: ')
    print(model.summary())

    # Train the model
    history = model.fit(train_x, train_y, verbose=0, batch_size=args.batch_size, epochs=args.
epochs, callbacks=[MetricsPrint()])

if __name__ == "__main__":
    main()

```

Metric Block

Content

소스 코드 상의 metric type

main 코드 블록 내에 metrics 항목으로 설정되어있습니다.

현재 제공하는 두 가지 예시 코드 모두 accuracy 로 설정하시면 됩니다.

```
model.compile(optimizer, loss='categorical_crossentropy', metrics=['accuracy'])
```

Input Example

```
accuracy
```