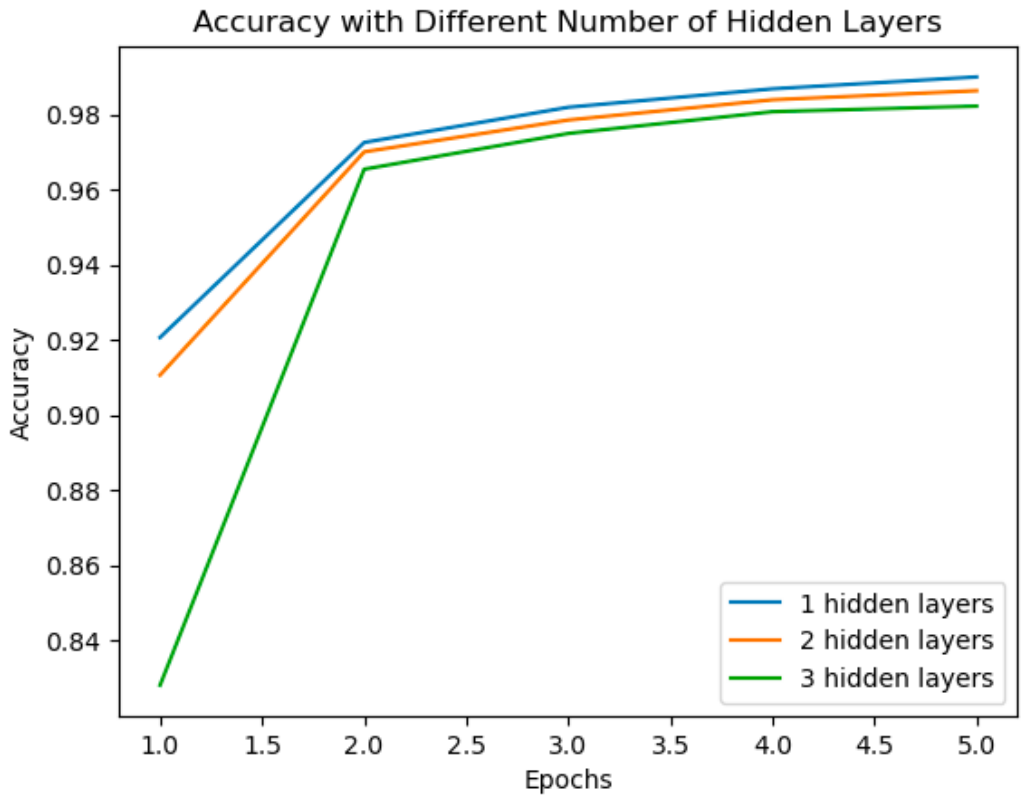
딥러닝입문
Assignment3

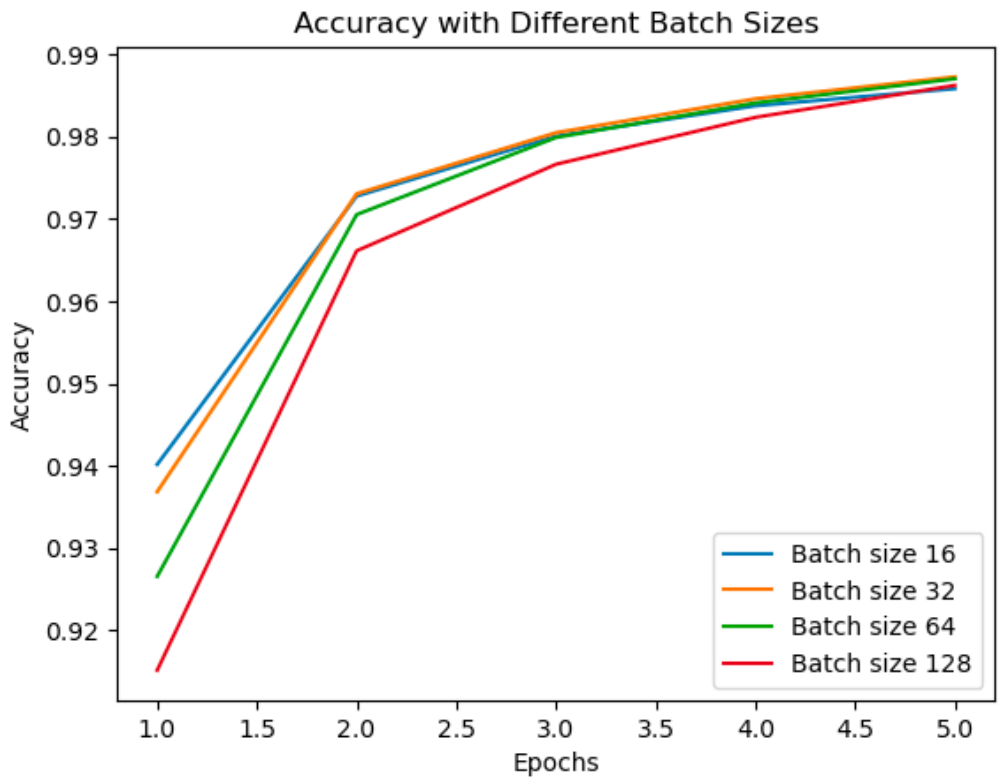
2021270660 이지원

① Plot the accuracy when you have [1,2,3] hidden layers in DNN model



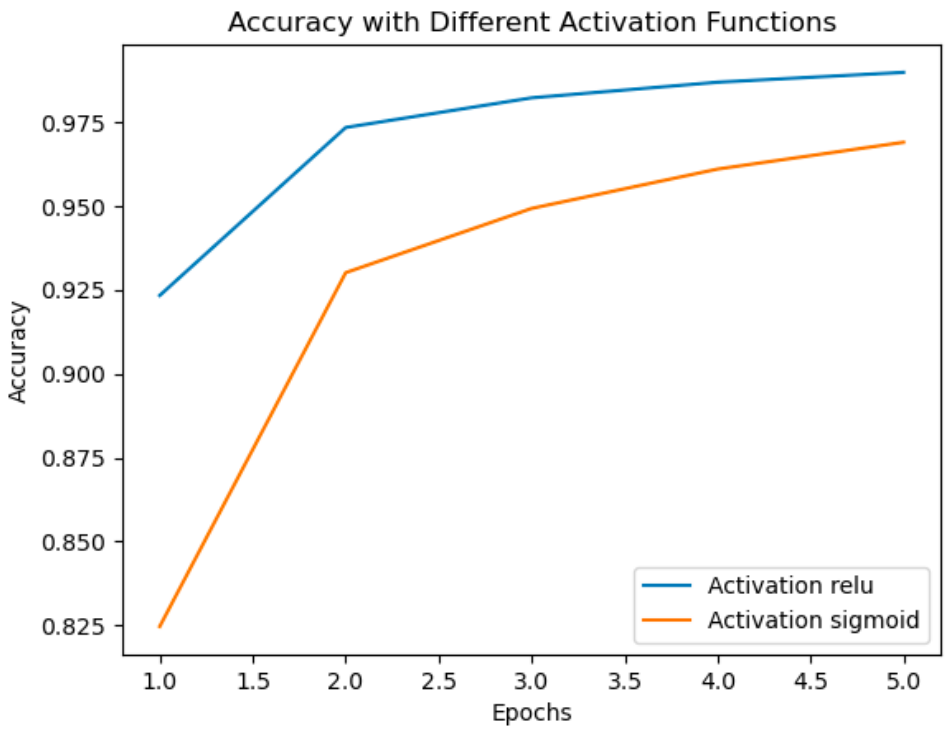

```python
import tensorflow as tf
import matplotlib.pyplot as plt

def build_and_train_model(hidden_layers, activation='relu', batch_size=128):
    model = tf.keras.models.Sequential()
    model.add(tf.keras.layers.Dense(512, activation=activation, input_shape=(784,)))
    for _ in range(hidden_layers):
        model.add(tf.keras.layers.Dense(512, activation=activation))
    model.add(tf.keras.layers.Dense(10, activation='sigmoid'))

    model.compile(optimizer='rmsprop', loss='mse', metrics=['accuracy'])
    history = model.fit(train_images, train_labels, epochs=5, batch_size=batch_size, verbose=0)
    return history.history['accuracy']

accuracies = []
for layers in [1, 2, 3]:
    acc = build_and_train_model(layers)
    accuracies.append(acc)

epochs = range(1, 6)
plt.figure()
for i, acc in enumerate(accuracies):
    plt.plot(epochs, acc, label=f'{i+1} hidden layers')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
plt.title('Accuracy with Different Number of Hidden Layers')
plt.show()
```

② Plot the accuracy when you change batchsize to 16, 32, 64, 128




```python
def build_and_train_model_with_batch_size(batch_size):
    model = tf.keras.models.Sequential()
    model.add(tf.keras.layers.Dense(512, activation='relu', input_shape=(784,)))
    model.add(tf.keras.layers.Dense(10, activation='sigmoid'))

    model.compile(optimizer='rmsprop', loss='mse', metrics=['accuracy'])
    history = model.fit(train_images, train_labels, epochs=5, batch_size=batch_size, verbose=0)
    return history.history['accuracy']

batch_sizes = [16, 32, 64, 128]
accuracies = []
for bs in batch_sizes:
    acc = build_and_train_model_with_batch_size(bs)
    accuracies.append(acc)

plt.figure()

for bs, acc in zip(batch_sizes, accuracies):
    plt.plot(epochs, acc, label=f'Batch size {bs}')

plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
plt.title('Accuracy with Different Batch Sizes')
plt.show()
```

③ Plot the accuracy when you change the activation function from ReLU to sigmoid




```python
def build_and_train_model_with_activation(activation):
    model = tf.keras.models.Sequential()
    model.add(tf.keras.layers.Dense(512, activation=activation, input_shape=(784,)))
    model.add(tf.keras.layers.Dense(512, activation=activation))
    model.add(tf.keras.layers.Dense(10, activation='sigmoid'))

    model.compile(optimizer='rmsprop', loss='mse', metrics=['accuracy'])
    history = model.fit(train_images, train_labels, epochs=5, batch_size=128, verbose=0)
    return history.history['accuracy']

activations = ['relu', 'sigmoid']
accuracies = []
for act in activations:
    acc = build_and_train_model_with_activation(act)
    accuracies.append(acc)

plt.figure()

for act, acc in zip(activations, accuracies):
    plt.plot(epochs, acc, label=f'Activation {act}')

plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
plt.title('Accuracy with Different Activation Functions')
plt.show()
```