Configuration	Hidden activation	Hidden Initializer	Output activation	Output initializer	Loss Function	Optimizer	Batch Size
1	tanh	Uniform 0.1	Sigmoid	Uniform 0.1	MSE	SGD Lr = 0.01	1
2	tanh	Uniform 0.1	Sigmoid	Uniform 0.1	MSE	SGD Lr = 10.0	1
3	tanh	Glorot uniform	Sigmoid	Glorot uniform	MSE	Adam	1
4	ReLU	He Normal	Softmax	Glorot uniform	CE	Adam	1
5	ReLU	He Normal	Softmax	Glorot uniform	CE	Adam	64

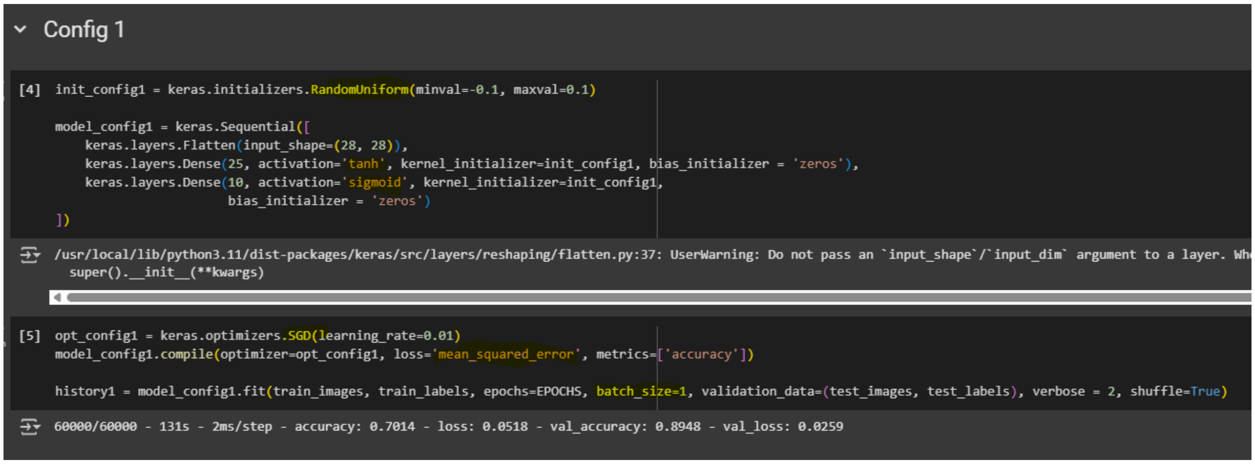
CE: Cross Entropy
MSE: Mean Squared Error
SGD: Stochastic Gradient Descent

Instructions: Refer to the table above for five different configurations of activation functions, initializers, hidden layers, output layers, loss functions, optimizers, and batch size. For this assignment, you should be referring to the lecture recordings in which I have coded configuration 1 as an example. Next, you need to experiment with the rest four configurations (from configuration 2 to 5) using keras API. For each configuration, you should show the following:

1. A snapshot showing all the parameters set per configuration. **(4 x 2.5 points)**
2. A snapshot of program run. **(4 x 2.5 points)**

NOTE: You can set the EPOCH to a value of one (1) to cut the processing time. You must provide a snapshot of parameters set and the program run for each configuration (from 2 to 5) to receive maximum points.

Submit your work in D2L before the due date.



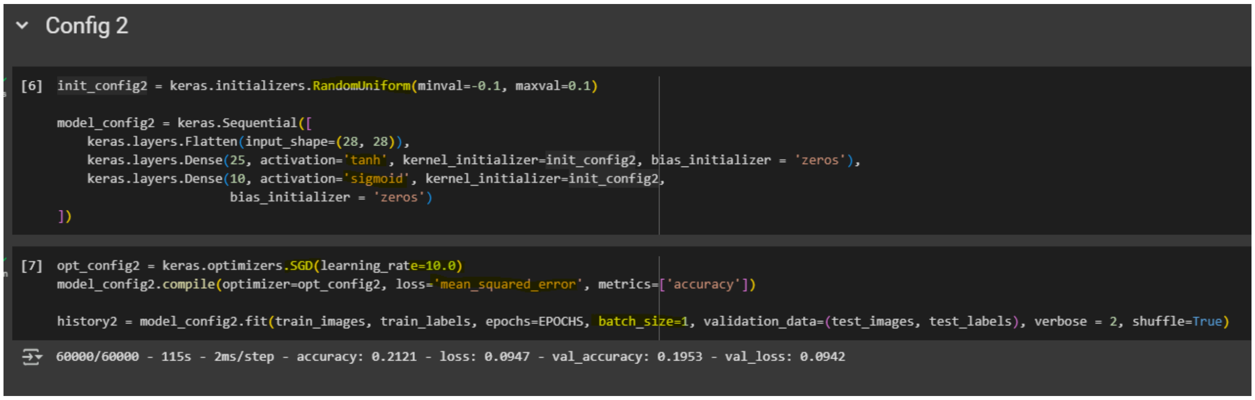
Config 1

```
[4] init_config1 = keras.initializers.RandomUniform(minval=-0.1, maxval=0.1)

    model_config1 = keras.Sequential([
        keras.layers.Flatten(input_shape=(28, 28)),
        keras.layers.Dense(25, activation='tanh', kernel_initializer=init_config1, bias_initializer = 'zeros'),
        keras.layers.Dense(10, activation='sigmoid', kernel_initializer=init_config1,
                           bias_initializer = 'zeros')
    ])
```

```
/usr/local/lib/python3.11/dist-packages/keras/src/layers/reshaping/flatten.py:37: UserWarning: Do not pass an `input_shape`/`input_dim` argument to a layer. Whe
  super().__init__(**kwargs)
```

```
[5] opt_config1 = keras.optimizers.SGD(learning_rate=0.01)
    model_config1.compile(optimizer=opt_config1, loss='mean_squared_error', metrics=['accuracy'])

    history1 = model_config1.fit(train_images, train_labels, epochs=EPOCHS, batch_size=1, validation_data=(test_images, test_labels), verbose = 2, shuffle=True)
```

```
60000/60000 - 131s - 2ms/step - accuracy: 0.7014 - loss: 0.0518 - val_accuracy: 0.8948 - val_loss: 0.0259
```

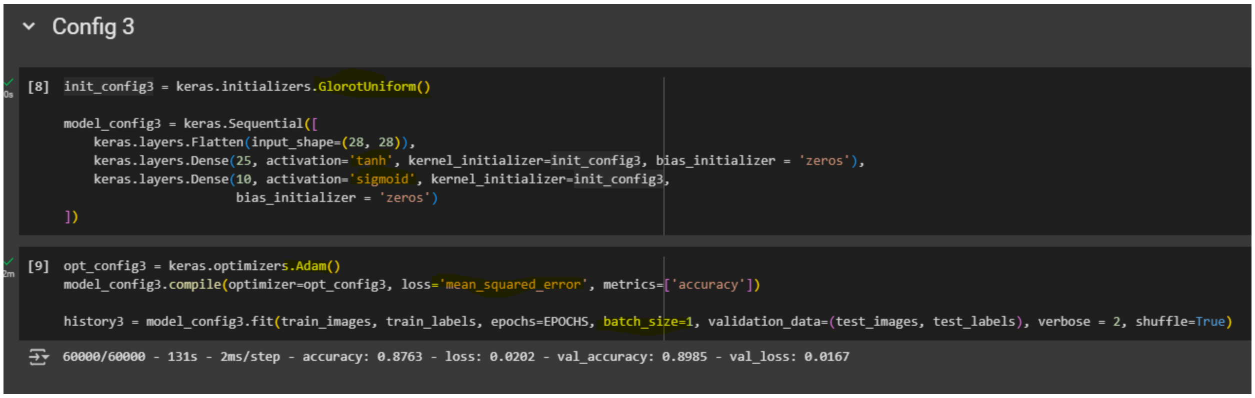


Config 2

```
[6] init_config2 = keras.initializers.RandomUniform(minval=-0.1, maxval=0.1)

    model_config2 = keras.Sequential([
        keras.layers.Flatten(input_shape=(28, 28)),
        keras.layers.Dense(25, activation='tanh', kernel_initializer=init_config2, bias_initializer = 'zeros'),
        keras.layers.Dense(10, activation='sigmoid', kernel_initializer=init_config2,
                           bias_initializer = 'zeros')
    ])
```

```
[7] opt_config2 = keras.optimizers.SGD(learning_rate=10.0)
    model_config2.compile(optimizer=opt_config2, loss='mean_squared_error', metrics=['accuracy'])

    history2 = model_config2.fit(train_images, train_labels, epochs=EPOCHS, batch_size=1, validation_data=(test_images, test_labels), verbose = 2, shuffle=True)
```

```
60000/60000 - 115s - 2ms/step - accuracy: 0.2121 - loss: 0.0947 - val_accuracy: 0.1953 - val_loss: 0.0942
```

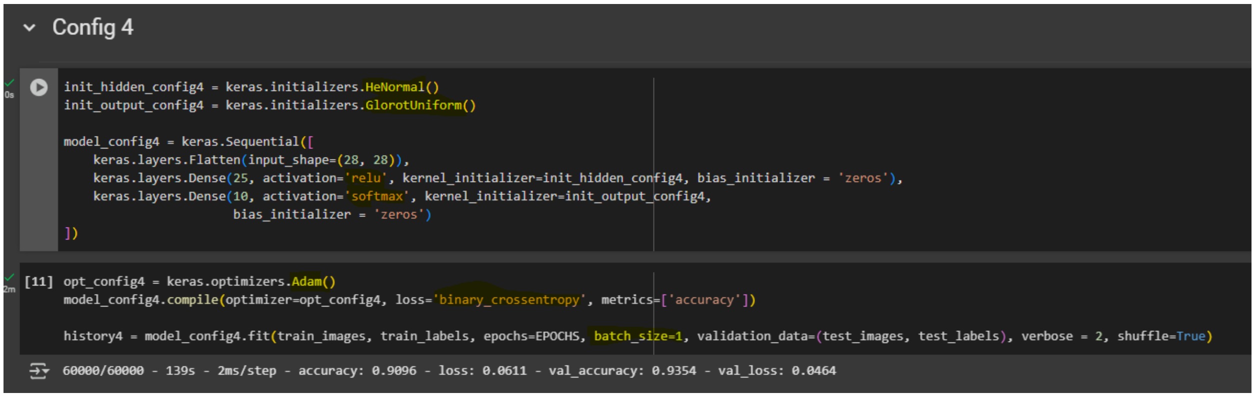


Config 3

```
[8] init_config3 = keras.initializers.GlorotUniform()

    model_config3 = keras.Sequential([
        keras.layers.Flatten(input_shape=(28, 28)),
        keras.layers.Dense(25, activation='tanh', kernel_initializer=init_config3, bias_initializer = 'zeros'),
        keras.layers.Dense(10, activation='sigmoid', kernel_initializer=init_config3,
                           bias_initializer = 'zeros')
    ])
```

```
[9] opt_config3 = keras.optimizers.Adam()
    model_config3.compile(optimizer=opt_config3, loss='mean_squared_error', metrics=['accuracy'])

    history3 = model_config3.fit(train_images, train_labels, epochs=EPOCHS, batch_size=1, validation_data=(test_images, test_labels), verbose = 2, shuffle=True)
```

```
60000/60000 - 131s - 2ms/step - accuracy: 0.8763 - loss: 0.0202 - val_accuracy: 0.8985 - val_loss: 0.0167
```

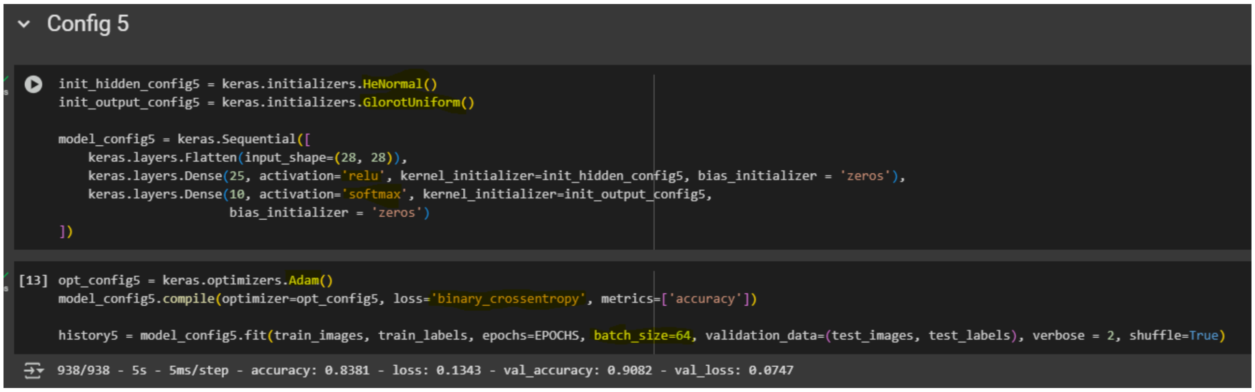


Config 4

```
init_hidden_config4 = keras.initializers.HeNormal()
init_output_config4 = keras.initializers.GlorotUniform()

model_config4 = keras.Sequential([
    keras.layers.Flatten(input_shape=(28, 28)),
    keras.layers.Dense(25, activation='relu', kernel_initializer=init_hidden_config4, bias_initializer = 'zeros'),
    keras.layers.Dense(10, activation='softmax', kernel_initializer=init_output_config4,
                       bias_initializer = 'zeros')
])
```

```
[11] opt_config4 = keras.optimizers.Adam()
     model_config4.compile(optimizer=opt_config4, loss='binary_crossentropy', metrics=['accuracy'])

     history4 = model_config4.fit(train_images, train_labels, epochs=EPOCHS, batch_size=1, validation_data=(test_images, test_labels), verbose = 2, shuffle=True)
```

```
60000/60000 - 139s - 2ms/step - accuracy: 0.9096 - loss: 0.0611 - val_accuracy: 0.9354 - val_loss: 0.0464
```



Config 5

```
init_hidden_config5 = keras.initializers.HeNormal()
init_output_config5 = keras.initializers.GlorotUniform()

model_config5 = keras.Sequential([
    keras.layers.Flatten(input_shape=(28, 28)),
    keras.layers.Dense(25, activation='relu', kernel_initializer=init_hidden_config5, bias_initializer = 'zeros'),
    keras.layers.Dense(10, activation='softmax', kernel_initializer=init_output_config5,
                       bias_initializer = 'zeros')
])
```

```
[13] opt_config5 = keras.optimizers.Adam()
     model_config5.compile(optimizer=opt_config5, loss='binary_crossentropy', metrics=['accuracy'])

     history5 = model_config5.fit(train_images, train_labels, epochs=EPOCHS, batch_size=64, validation_data=(test_images, test_labels), verbose = 2, shuffle=True)
```

```
938/938 - 5s - 5ms/step - accuracy: 0.8381 - loss: 0.1343 - val_accuracy: 0.9082 - val_loss: 0.0747
```