

Hyper-parameters:

The data is already splitted. For the input and hidden layer, we use the TANH as an activation function, and the last layer uses the SOFTMAX. The used optimizer is ADAM, and for loss function, my choice was sparse_categorical_crossentropy. The metric is accuracy. The number of epochs is 3. The batch_size value is 40.

Model:

The model is Sequential. I used 3 layers. The first one is the input layer. The second is the hidden layer. This layer consists of 32 neurons. The last layer is the output, and the number of neurons is 10.

```
print("--Make model--")
model = tf.keras.models.Sequential([
    tf.keras.layers.Flatten(input_shape=(28, 28)),
    tf.keras.layers.Dense(32, activation='tanh'),
    tf.keras.layers.Dense(10, activation='softmax')
])
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])

print("--Fit model--")
model.fit(x_train, y_train, epochs=3, batch_size=40, verbose=2)
```

Results:

The achieved accuracy for trained data is 96.1%. The accuracy for test data is 95.4%.

The changes:

I added 1 more layer to the model with the TANH activation function. The ADAM optimizer leads to a higher result in comparison with the SGD optimizer. I increased the number of epochs to give the model the chance to go over the network more than one time and learn from the previous mistakes. Moreover, I introduced a new parameter called batch_size and its value is 40.