kindly adjust the mnist dataset model architecture trying to improve the accuracy

and give me brief description about the best parameters

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **layers** | **BATCH\_SIZE** | **EPOCHS** | **OPTIMISER** | **normalization** | **validation accuracy** | **testing accuracy** | **val\_loss** |
| 1 | 10,10 | 128 | 28 | ADAM | without batch normalization | 0.9438 | 0.9319 | 0.1993 |
| 2 | 10,10 | 128 | 28 | ADAM | without batch normalization | 0.9448 | 0.9309 | 0.1922 |
| 3 | 10,10 | 128 | 28 | ADAM | with batch normalization | 0.9408 | 0.9307 | 0.2047 |
| 4 | 32,64,32,10 | 128 | 28 | ADAM | without batch normalization | 0.9718 | 0.9697 | 0.1286 |
| 5 | 32,64,32,10 | 128 | 28 | ADAM | without batch normalization | 0.9757 | 0.9709 | 0.1263 |
| 6 | 32,64,32,10 | 128 | 28 | ADAM | without batch normalization | 0.9737 | 0.9699 | 0.1274 |
| 7 | 32,64,32,10 | 128 | 28 | ADAM | with batch normalization | 0.9753 | 0.9723 | 0.1113 |
| 8 | 32,64,32,10 | 128 | 28 | ADAM | with batch normalization | 0.9728 | 0.9719 | 0.1168 |
| 9 | 32,64,32,10 | 128 | 28 | SGD | with batch normalization | 0.976 | 0.9714 | 0.099 |
| 10 | 32,64,32,10 | 128 | 28 | SGD | with batch normalization | 0.9772 | 0.9741 | 0.0955 |

and regarding for your solution please answer below questions

Question 1: Why is it necessary to normalize the data before training a neural network?

when input features have different scales (e.g., one feature ranges from 0 to 1, while another ranges from 1000 to 10000), it can make the gradient descent process slower and inefficient. Large variations in input values can cause the optimizer to take very small steps in some directions and very large steps in others, leading to inefficient training.

Question .2: Explain the purpose of reshaping the images to a flat vector using reshape(-1, 784) in the code.

We are conversting 2D image data (28x28) to 1D vector 784 Because in neural network the input is on 1D vector shape not 2D

What is batch normalization, and why is it used? Describe how batch normalization affects the learning process.

Batch normalization is a technique used in neural networks to normalize before getting into activation step, It helps stabilize and accelerate the training process by ensuring that the inputs to each layer are normalized to have a consistent distribution.

Task 3:

Define a simple feedforward neural network using tf.keras.Sequential as in the code. The model should contain:

A fully connected Dense layer with 10 units.

Batch Normalization after the Dense layer.

An activation function (tanh) applied after batch normalization.

A final Dense layer with 10 units and a softmax activation function for multi-class classification.

How does batch normalization help with training deep neural networks? Provide your thoughts after training the model.

Before batch normalization the result was

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **layers** | **BATCH\_SIZE** | **EPOCHS** | **OPTIMISER** | **activation** | **normalization** | **validation accuracy** | **testing accuracy** | **val\_loss** |
| 10,10 | 128 | 28 | ADAM | tanh | without batch normalization | 0.9438 | 0.9319 | 0.1993 |

After batch normalization the result was

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **layers** | **BATCH\_SIZE** | **EPOCHS** | **OPTIMISER** | **activation** | **normalization** | **validation accuracy** | **testing accuracy** | **val\_loss** |
| 10,10 | 128 | 28 | ADAM | tanh | with batch normalization | 0.9408 | 0.9307 | 0.2047 |

Batch normalization made the results worst in this case

But when I increased layers and tried batch normalization it made a little better difference

How does the choice of activation function (e.g., tanh, relu, sigmoid) affect model performance? Experiment with changing the activation function

for every question need specific answer and prove it on your model result

Relu was better when I tired it compared to tanh in this case