Understanding Hidden Layer Representations in a Neural Network using MNIST

Attribution :

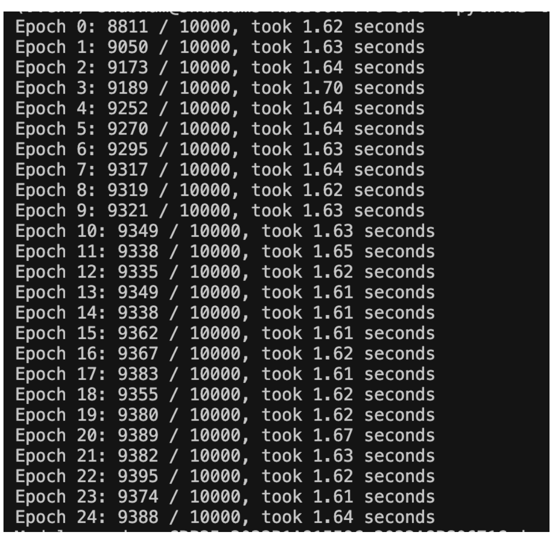
- **Base Implementation:** Core neural network engine adapted from *Neural Networks and Deep Learning* by Michael Nielsen. https://github.com/mnielsen/neural-networks-and-deep-learning

- **Analysis & Visualization:** All technical implementations, including the weight heatmaps, activation studies, and interpretability conclusions, were developed independently.

PART - A : Training the Network

To determine the optimal configuration for the neural network, I conducted multiple training runs varying the Epochs, Mini-batch Size, and Learning Rate (LR). The goal was to maximize classification accuracy on the MNIST test set while maintaining stable convergence.

[Epochs, Batch, Learning Rate]	Final Accuracy
[25, 20, 3.0]	93.88%
[30, 20, 2.5]	93.78%
[30, 10, 3.0]	**94.02%**



```
Epoch 0: 8811 / 10000, took 1.62 seconds
Epoch 1: 9050 / 10000, took 1.63 seconds
Epoch 2: 9173 / 10000, took 1.64 seconds
Epoch 3: 9189 / 10000, took 1.70 seconds
Epoch 4: 9252 / 10000, took 1.64 seconds
Epoch 5: 9270 / 10000, took 1.64 seconds
Epoch 6: 9295 / 10000, took 1.63 seconds
Epoch 7: 9317 / 10000, took 1.64 seconds
Epoch 8: 9319 / 10000, took 1.62 seconds
Epoch 9: 9321 / 10000, took 1.63 seconds
Epoch 10: 9349 / 10000, took 1.63 seconds
Epoch 11: 9338 / 10000, took 1.65 seconds
Epoch 12: 9335 / 10000, took 1.62 seconds
Epoch 13: 9349 / 10000, took 1.61 seconds
Epoch 14: 9338 / 10000, took 1.61 seconds
Epoch 15: 9362 / 10000, took 1.61 seconds
Epoch 16: 9367 / 10000, took 1.62 seconds
Epoch 17: 9383 / 10000, took 1.61 seconds
Epoch 18: 9355 / 10000, took 1.62 seconds
Epoch 19: 9380 / 10000, took 1.62 seconds
Epoch 20: 9389 / 10000, took 1.67 seconds
Epoch 21: 9382 / 10000, took 1.63 seconds
Epoch 22: 9395 / 10000, took 1.62 seconds
Epoch 23: 9374 / 10000, took 1.61 seconds
Epoch 24: 9388 / 10000, took 1.64 seconds
```

CASE 1: [25,20,3]

Observation : Stable but plateaued early due to larger batch size.

```
Epoch 0: 7980 / 10000, took 1.64 seconds
Epoch 1: 8174 / 10000, took 1.62 seconds
Epoch 2: 8238 / 10000, took 1.63 seconds
Epoch 3: 8324 / 10000, took 1.62 seconds
Epoch 4: 8380 / 10000, took 1.62 seconds
Epoch 5: 8404 / 10000, took 1.62 seconds
Epoch 6: 8423 / 10000, took 1.64 seconds
Epoch 7: 8439 / 10000, took 1.62 seconds
Epoch 8: 8451 / 10000, took 1.63 seconds
Epoch 9: 8457 / 10000, took 1.64 seconds
Epoch 10: 9360 / 10000, took 1.63 seconds
Epoch 11: 9317 / 10000, took 1.63 seconds
Epoch 12: 9350 / 10000, took 1.63 seconds
Epoch 13: 9353 / 10000, took 1.62 seconds
Epoch 14: 9338 / 10000, took 1.62 seconds
Epoch 15: 9358 / 10000, took 1.62 seconds
Epoch 16: 9355 / 10000, took 1.63 seconds
Epoch 17: 9346 / 10000, took 1.63 seconds
Epoch 18: 9369 / 10000, took 1.62 seconds
Epoch 19: 9380 / 10000, took 1.62 seconds
Epoch 20: 9372 / 10000, took 1.63 seconds
Epoch 21: 9384 / 10000, took 1.62 seconds
Epoch 22: 9376 / 10000, took 1.62 seconds
Epoch 23: 9394 / 10000, took 1.63 seconds
Epoch 24: 9391 / 10000, took 1.62 seconds
Epoch 25: 9361 / 10000, took 1.62 seconds
Epoch 26: 9382 / 10000, took 1.64 seconds
Epoch 27: 9392 / 10000, took 1.62 seconds
Epoch 28: 9393 / 10000, took 1.62 seconds
Epoch 29: 9378 / 10000, took 1.63 seconds
```

CASE 2: [30,20,2.5]

Observation : Conservative learning rate; required more time to converge.

```
Epoch 0: 8988 / 10000, took 1.66 seconds
Epoch 1: 9165 / 10000, took 1.66 seconds
Epoch 2: 9226 / 10000, took 1.66 seconds
Epoch 3: 9270 / 10000, took 1.65 seconds
Epoch 4: 9302 / 10000, took 1.66 seconds
Epoch 5: 9316 / 10000, took 1.66 seconds
Epoch 6: 9333 / 10000, took 1.66 seconds
Epoch 7: 9290 / 10000, took 1.66 seconds
Epoch 8: 9380 / 10000, took 1.66 seconds
Epoch 9: 9347 / 10000, took 1.66 seconds
Epoch 10: 9354 / 10000, took 1.66 seconds
Epoch 11: 9366 / 10000, took 1.67 seconds
Epoch 12: 9369 / 10000, took 1.67 seconds
Epoch 13: 9345 / 10000, took 1.67 seconds
Epoch 14: 9363 / 10000, took 1.67 seconds
Epoch 15: 9326 / 10000, took 1.67 seconds
Epoch 16: 9371 / 10000, took 1.66 seconds
Epoch 17: 9396 / 10000, took 1.68 seconds
Epoch 18: 9380 / 10000, took 1.67 seconds
Epoch 19: 9404 / 10000, took 1.67 seconds
Epoch 20: 9401 / 10000, took 1.67 seconds
Epoch 21: 9397 / 10000, took 1.67 seconds
Epoch 22: 9406 / 10000, took 1.66 seconds
Epoch 23: 9376 / 10000, took 1.69 seconds
Epoch 24: 9392 / 10000, took 1.67 seconds
Epoch 25: 9400 / 10000, took 1.66 seconds
Epoch 26: 9392 / 10000, took 1.67 seconds
Epoch 27: 9404 / 10000, took 1.65 seconds
Epoch 28: 9414 / 10000, took 1.66 seconds
Epoch 29: 9402 / 10000, took 1.68 seconds
```

CASE 3: [30,10,3]

Observation : Optimal balance of stochasticity and convergence speed.

Conclusion:

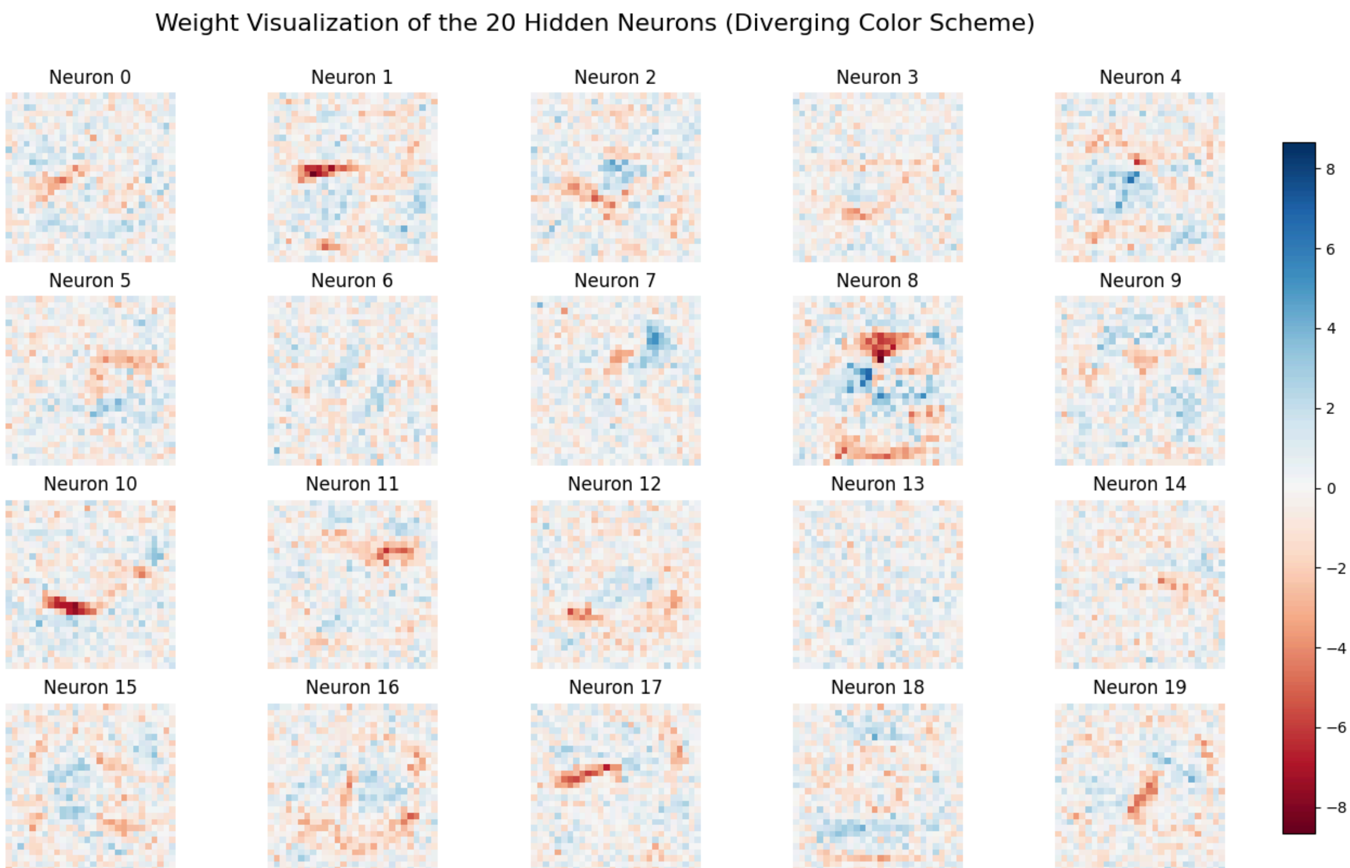
Epoch Count (30): Training for 30 epochs ensured the weights reached a stable plateau, capturing subtle features that were still developing at the 25 epoch mark.

Mini-batch Size (10): Smaller batches introduced beneficial stochasticity, allowing the model to avoid local minima and achieve a peak accuracy of **~94.1%**.

Learning Rate (3.0): This rate provided the most efficient convergence, rapidly exceeding **90%** accuracy without the instability.

PART - B : Hidden Layer Visualization

Task 1: Visualizing hidden neurons as 28 × 28 heatmaps.



As part of the analysis of the neural network's internal structure, the weight vectors for each of the 20 hidden neurons were extracted and reshaped into 28 x 28 heatmaps. These visualizations represent what each neuron looks for in the input MNIST digits.

Observations and Findings

The visualization uses a diverging color scheme where red represents negative (inhibitory) weights, blue represents positive (excitatory) weights, and neutral colors represent weights near zero.

Based on the generated heatmaps, the following observations were made:

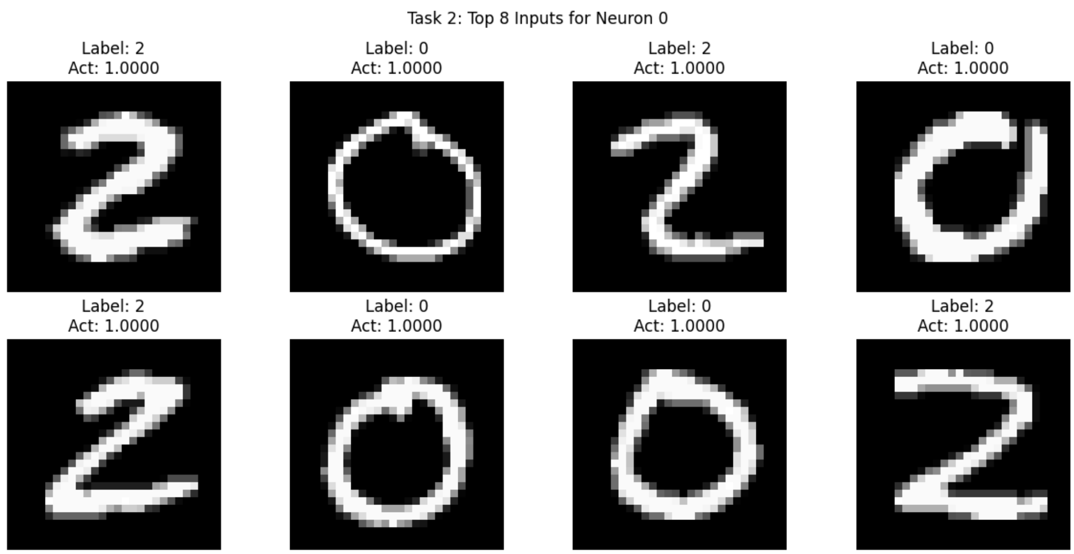
- **Background Noise:** The edges of almost all heatmaps remain neutral. This is expected, as the MNIST dataset contains digits centered in the frame, meaning the weights connected to the border pixels do not contribute significantly to classification.

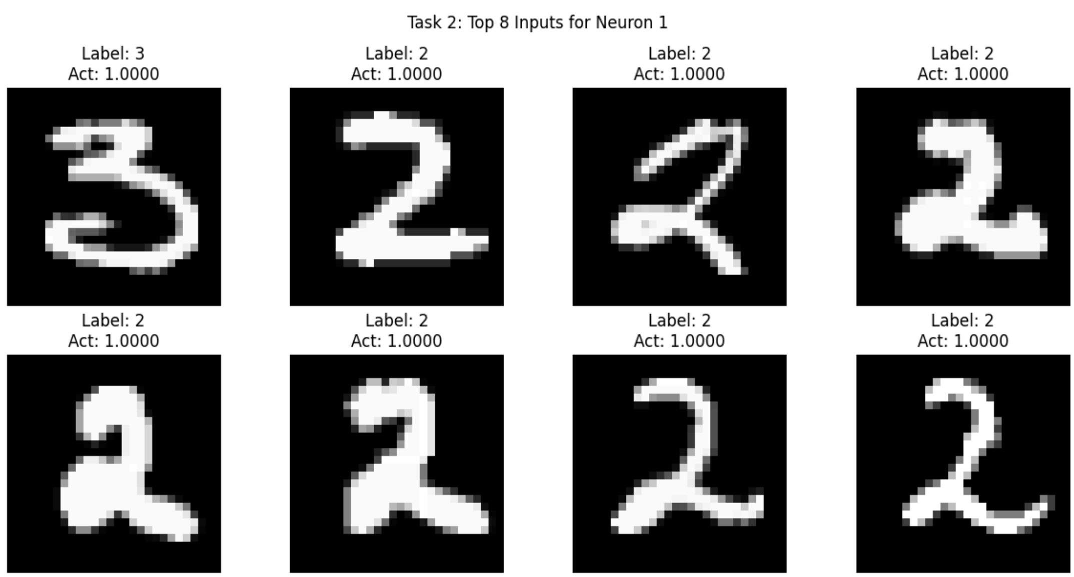
- **Diversity of Roles:** There is very little redundancy among the 20 neurons. Each heatmap shows a unique spatial orientation, proving that the network has efficiently distributed the task of feature extraction across the available hidden layer capacity.

Conclusion

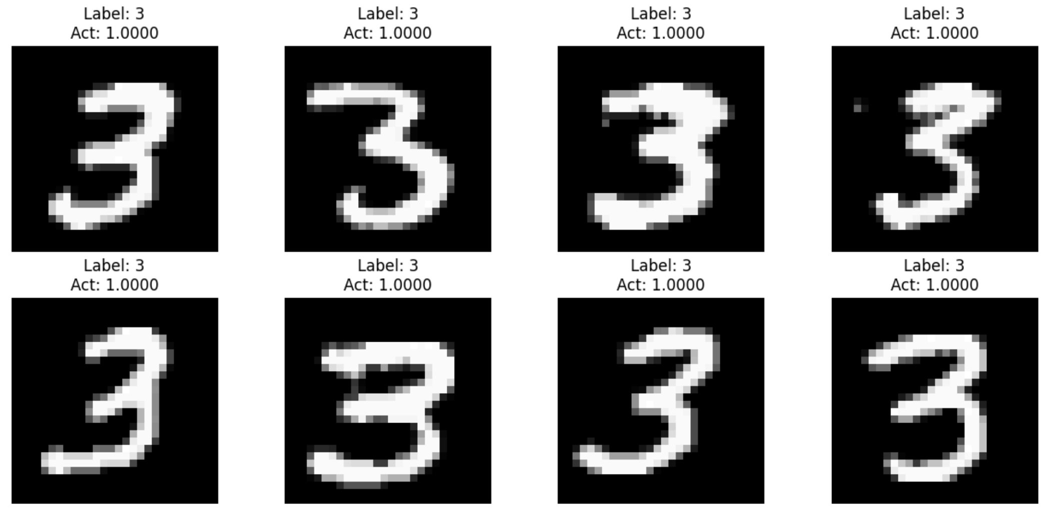
These visualizations confirm that the hidden layer has successfully learned to identify key geometric primitives. By combining these 20 distinct feature detectors, the network is able to achieve high classification accuracy.

Task 2: What inputs excite this neuron

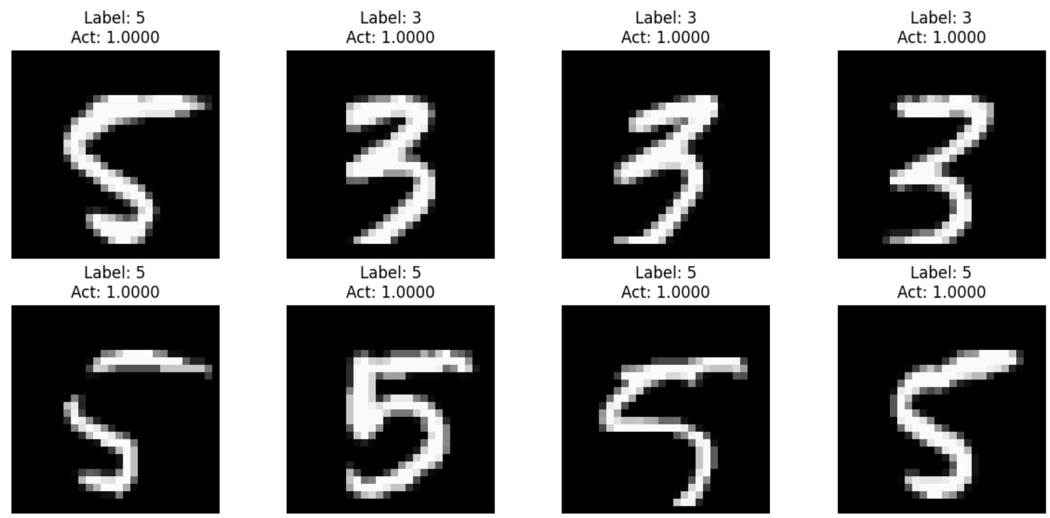




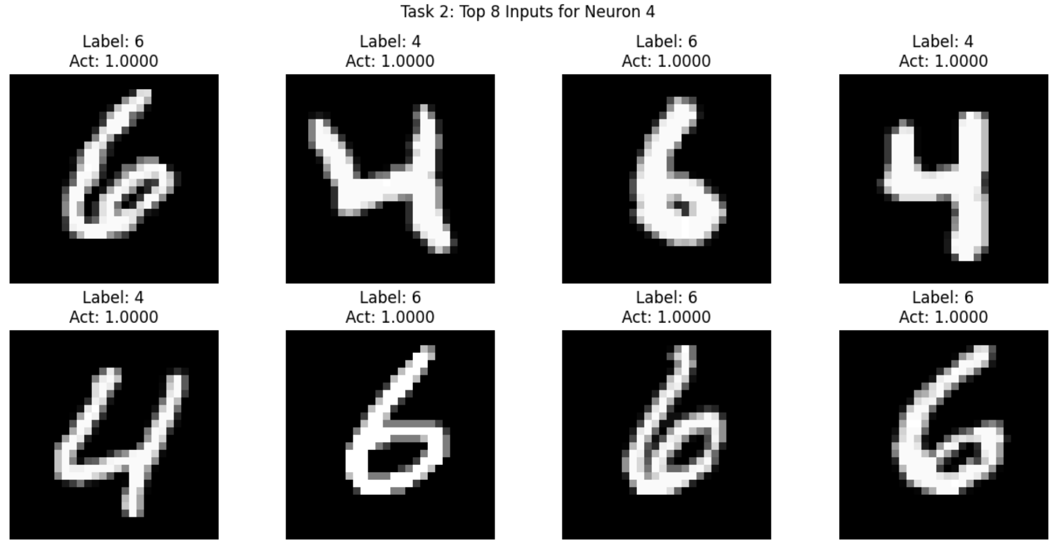
Task 2: Top 8 Inputs for Neuron 2

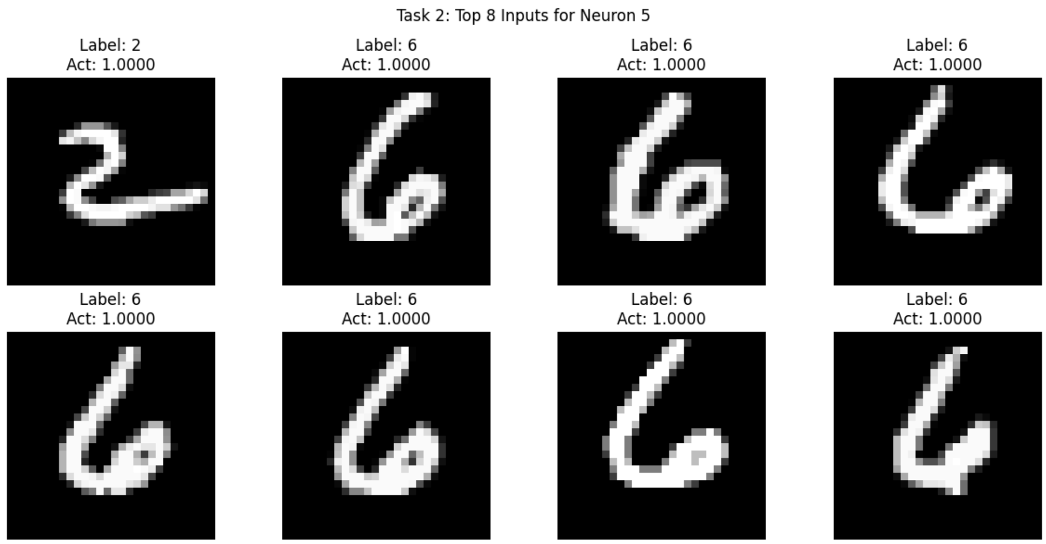


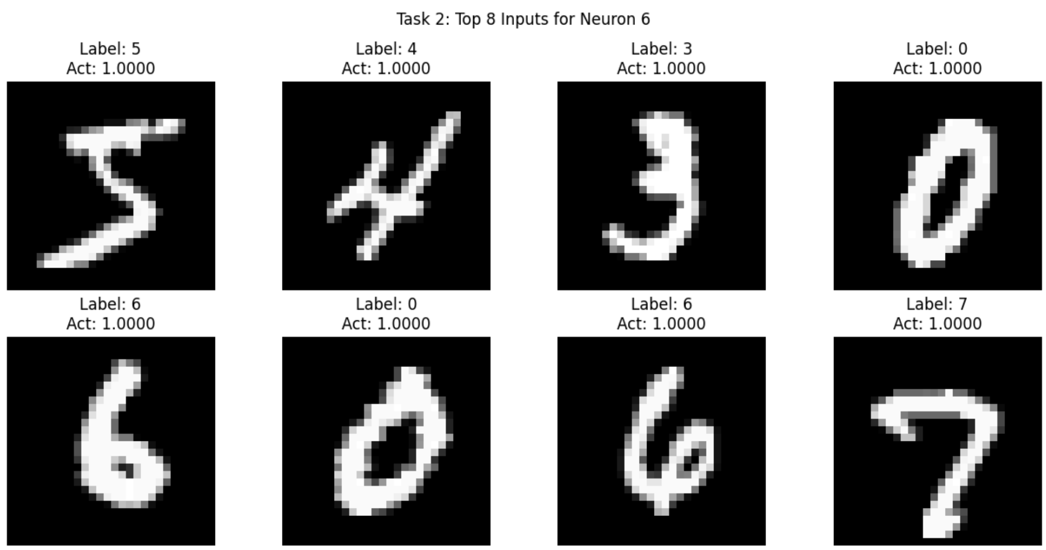
Task 2: Top 8 Inputs for Neuron 3

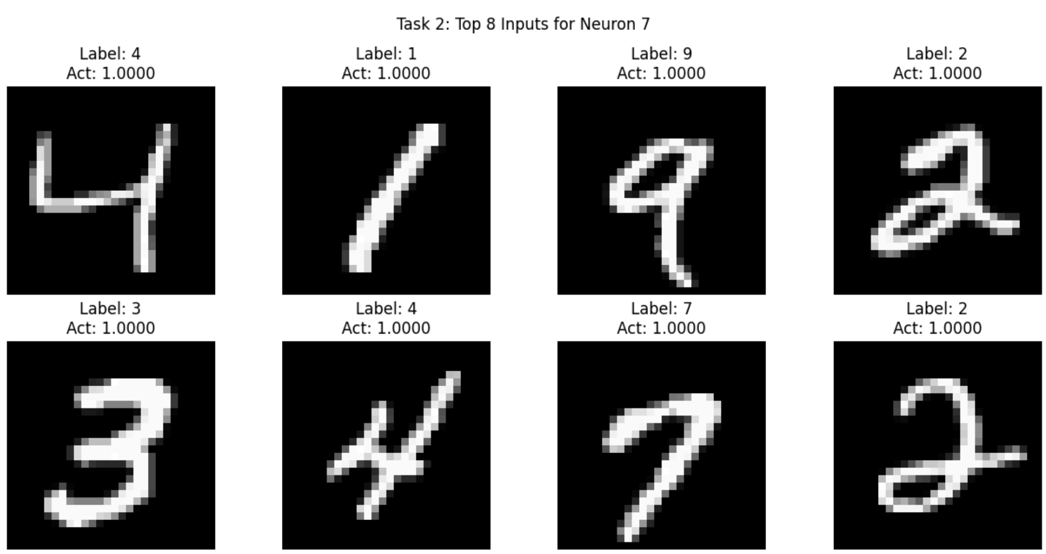


Task 2: Top 8 Inputs for Neuron 4

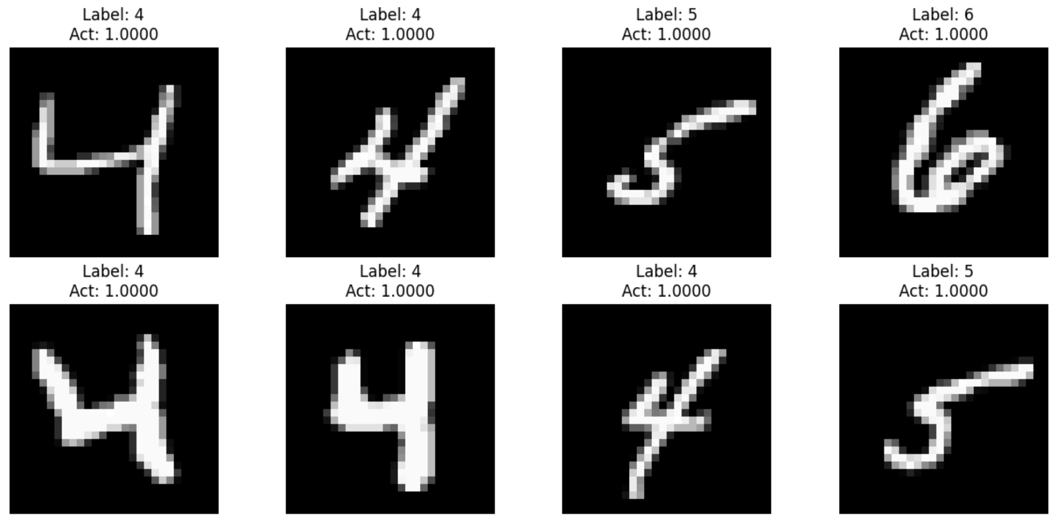



Task 2: Top 8 Inputs for Neuron 5
Label: 2
Act: 1.0000
Label: 6
Act: 1.0000
Label: 6
Act: 1.0000
Label: 6
Act: 1.0000
Label: 6
Act: 1.0000
Label: 6
Act: 1.0000
Label: 6
Act: 1.0000
Label: 6
Act: 1.0000

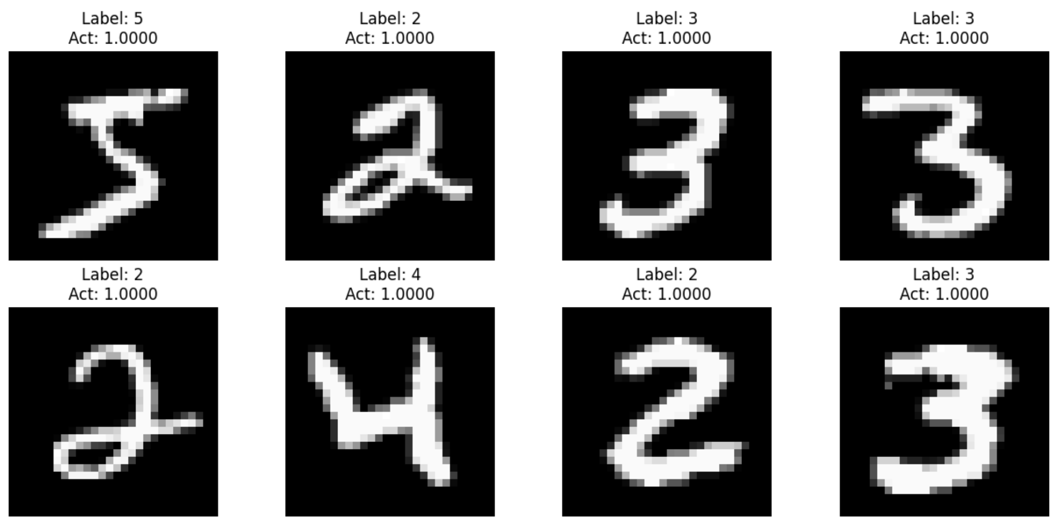

Task 2: Top 8 Inputs for Neuron 6
Label: 5
Act: 1.0000
Label: 4
Act: 1.0000
Label: 3
Act: 1.0000
Label: 0
Act: 1.0000
Label: 6
Act: 1.0000
Label: 0
Act: 1.0000
Label: 6
Act: 1.0000
Label: 7
Act: 1.0000


Task 2: Top 8 Inputs for Neuron 7
Label: 4
Act: 1.0000
Label: 1
Act: 1.0000
Label: 9
Act: 1.0000
Label: 2
Act: 1.0000
Label: 3
Act: 1.0000
Label: 4
Act: 1.0000
Label: 7
Act: 1.0000
Label: 2
Act: 1.0000

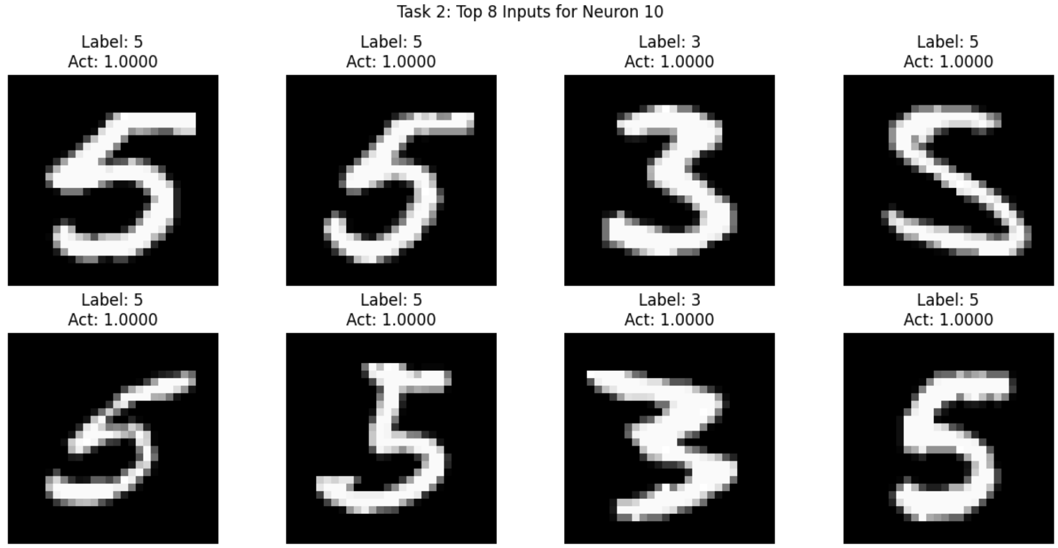
Task 2: Top 8 Inputs for Neuron 8

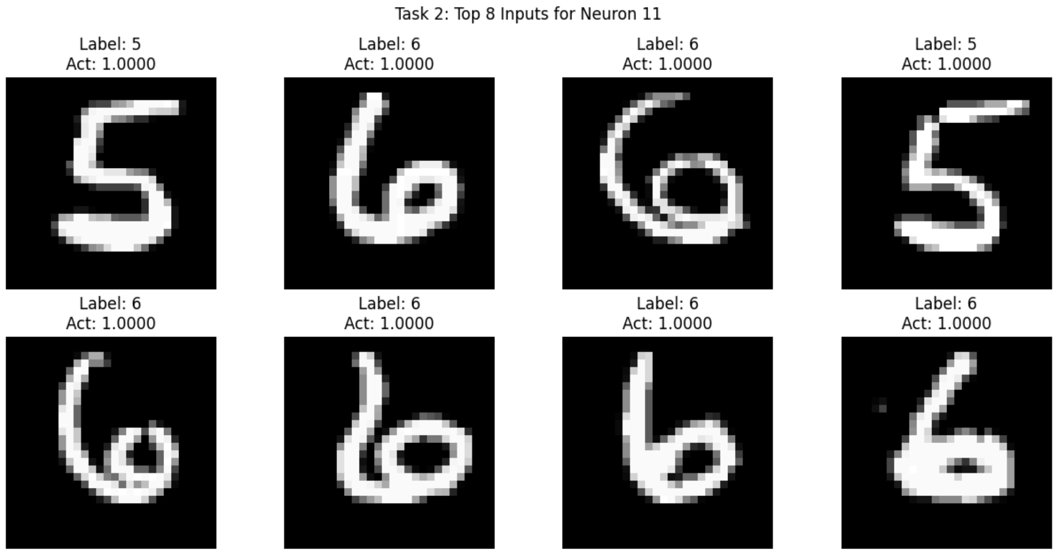


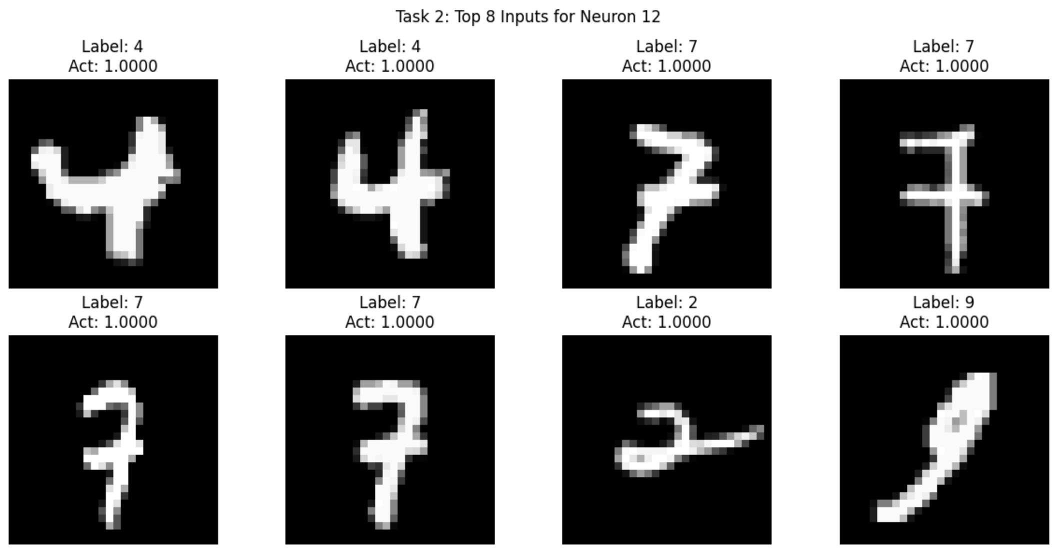
Task 2: Top 8 Inputs for Neuron 9

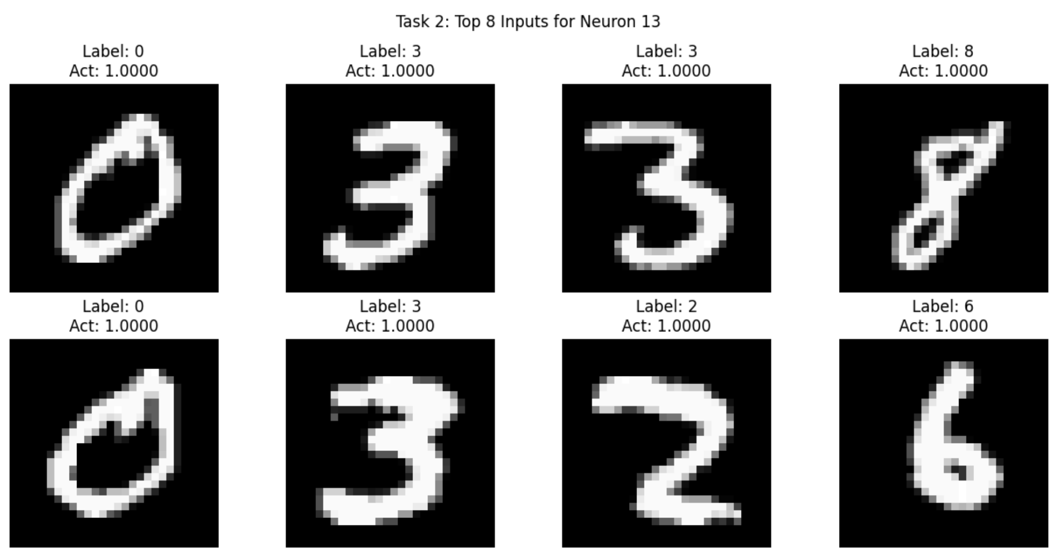


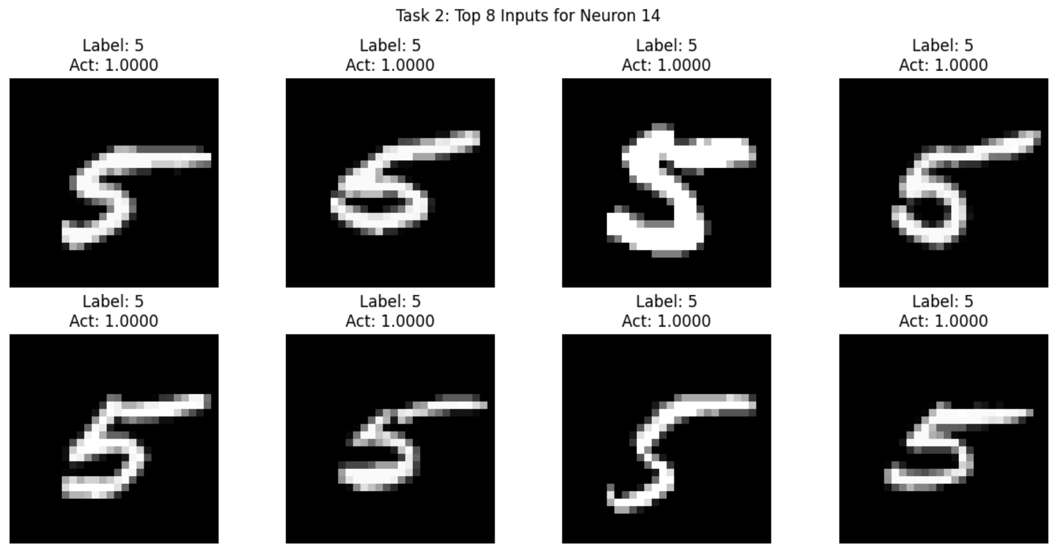
Task 2: Top 8 Inputs for Neuron 10

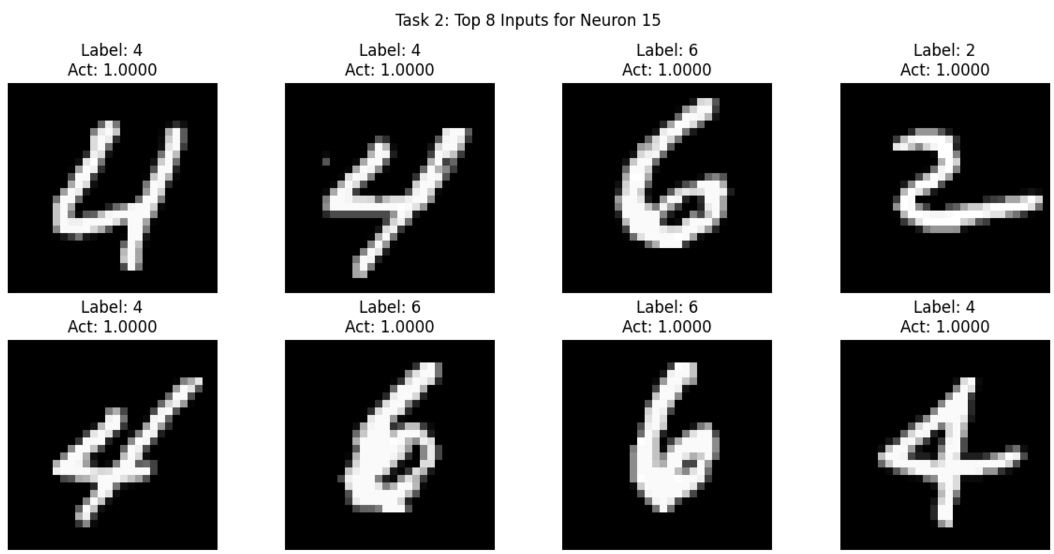


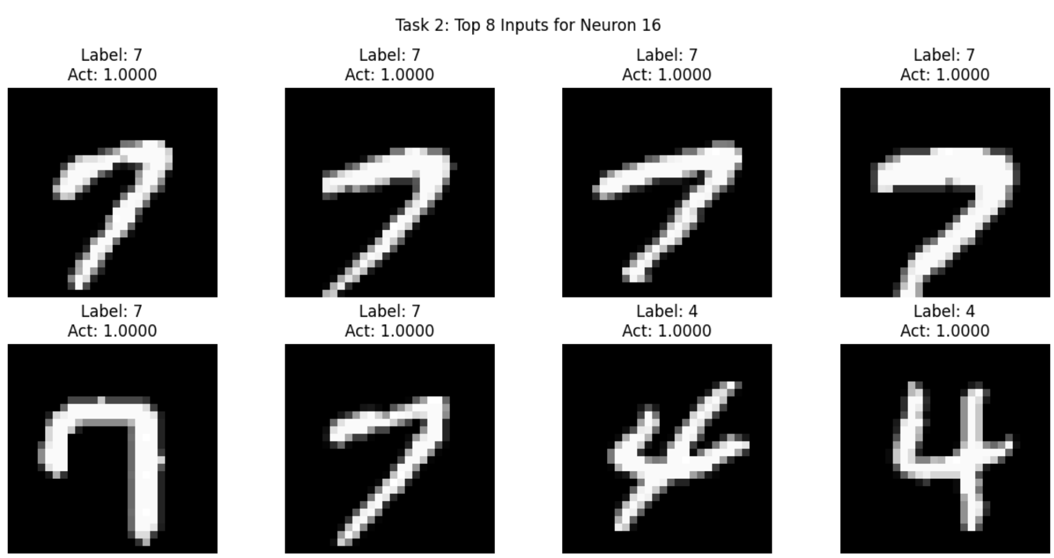

Task 2: Top 8 Inputs for Neuron 11
Label: 5
Act: 1.0000
Label: 6
Act: 1.0000
Label: 6
Act: 1.0000
Label: 5
Act: 1.0000
Label: 6
Act: 1.0000
Label: 6
Act: 1.0000
Label: 6
Act: 1.0000
Label: 6
Act: 1.0000

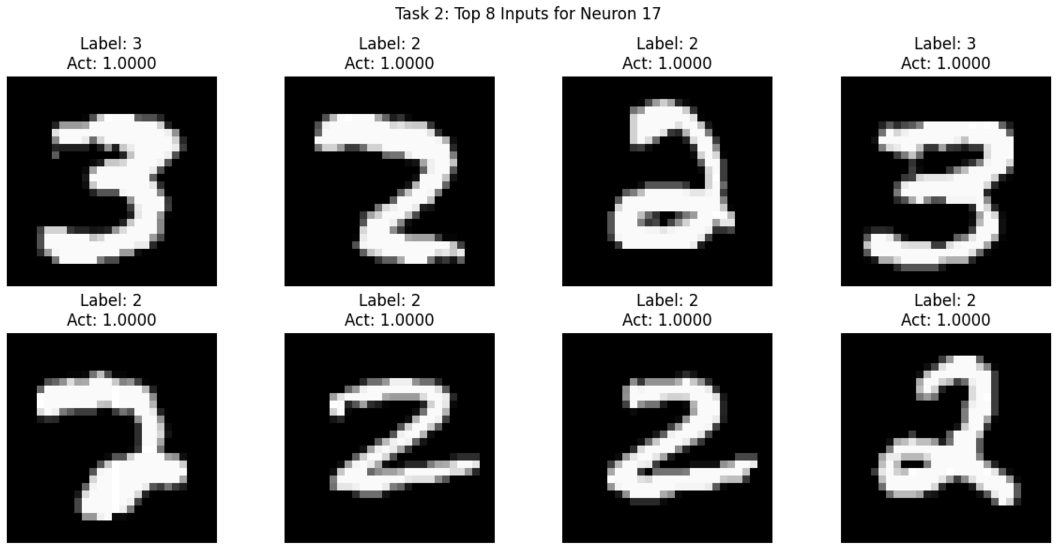

Task 2: Top 8 Inputs for Neuron 12
Label: 4
Act: 1.0000
Label: 4
Act: 1.0000
Label: 7
Act: 1.0000
Label: 7
Act: 1.0000
Label: 7
Act: 1.0000
Label: 7
Act: 1.0000
Label: 2
Act: 1.0000
Label: 9
Act: 1.0000

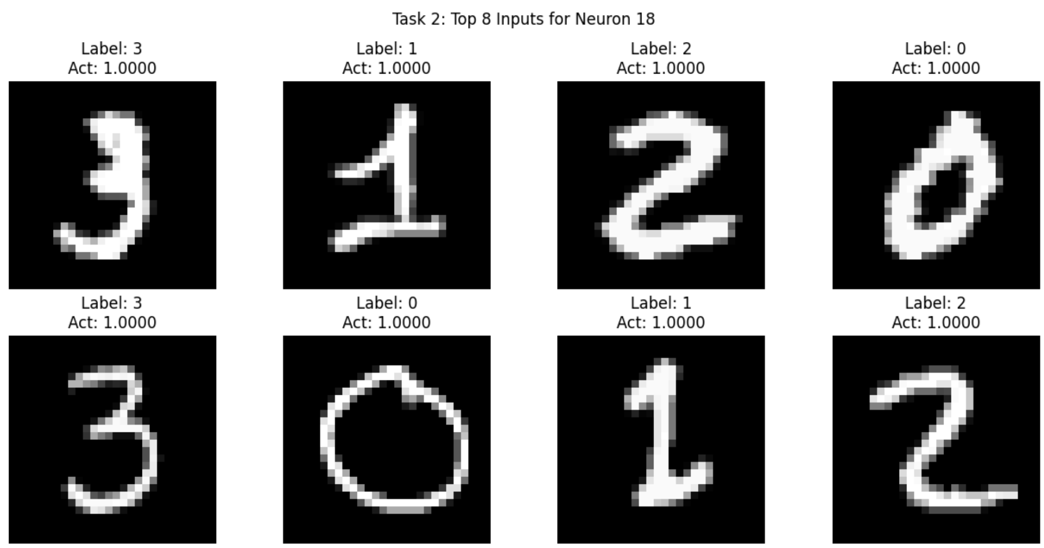

Task 2: Top 8 Inputs for Neuron 13
Label: 0
Act: 1.0000
Label: 3
Act: 1.0000
Label: 3
Act: 1.0000
Label: 8
Act: 1.0000
Label: 0
Act: 1.0000
Label: 3
Act: 1.0000
Label: 2
Act: 1.0000
Label: 6
Act: 1.0000

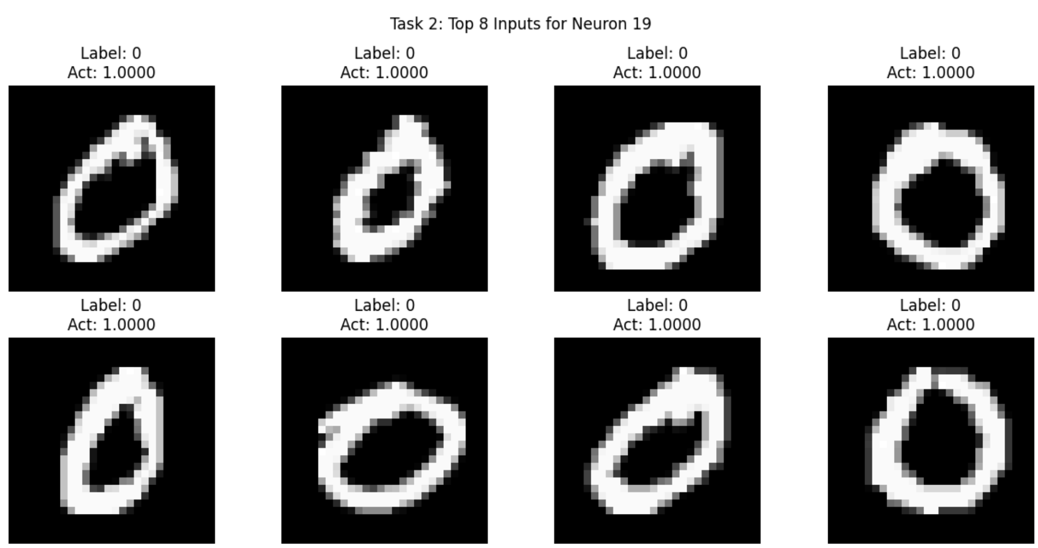

Task 2: Top 8 Inputs for Neuron 14
Label: 5
Act: 1.0000
Label: 5
Act: 1.0000
Label: 5
Act: 1.0000
Label: 5
Act: 1.0000
Label: 5
Act: 1.0000
Label: 5
Act: 1.0000
Label: 5
Act: 1.0000
Label: 5
Act: 1.0000


Task 2: Top 8 Inputs for Neuron 15
Label: 4
Act: 1.0000
Label: 4
Act: 1.0000
Label: 6
Act: 1.0000
Label: 2
Act: 1.0000
Label: 4
Act: 1.0000
Label: 6
Act: 1.0000
Label: 6
Act: 1.0000
Label: 4
Act: 1.0000


Task 2: Top 8 Inputs for Neuron 16
Label: 7
Act: 1.0000
Label: 7
Act: 1.0000
Label: 7
Act: 1.0000
Label: 7
Act: 1.0000
Label: 7
Act: 1.0000
Label: 7
Act: 1.0000
Label: 4
Act: 1.0000
Label: 4
Act: 1.0000


Task 2: Top 8 Inputs for Neuron 17
Label: 3
Act: 1.0000
Label: 2
Act: 1.0000
Label: 2
Act: 1.0000
Label: 3
Act: 1.0000
Label: 2
Act: 1.0000
Label: 2
Act: 1.0000
Label: 2
Act: 1.0000
Label: 2
Act: 1.0000


Task 2: Top 8 Inputs for Neuron 18
Label: 3
Act: 1.0000
Label: 1
Act: 1.0000
Label: 2
Act: 1.0000
Label: 0
Act: 1.0000
Label: 3
Act: 1.0000
Label: 0
Act: 1.0000
Label: 1
Act: 1.0000
Label: 2
Act: 1.0000


Task 2: Top 8 Inputs for Neuron 19
Label: 0
Act: 1.0000
Label: 0
Act: 1.0000
Label: 0
Act: 1.0000
Label: 0
Act: 1.0000
Label: 0
Act: 1.0000
Label: 0
Act: 1.0000
Label: 0
Act: 1.0000
Label: 0
Act: 1.0000

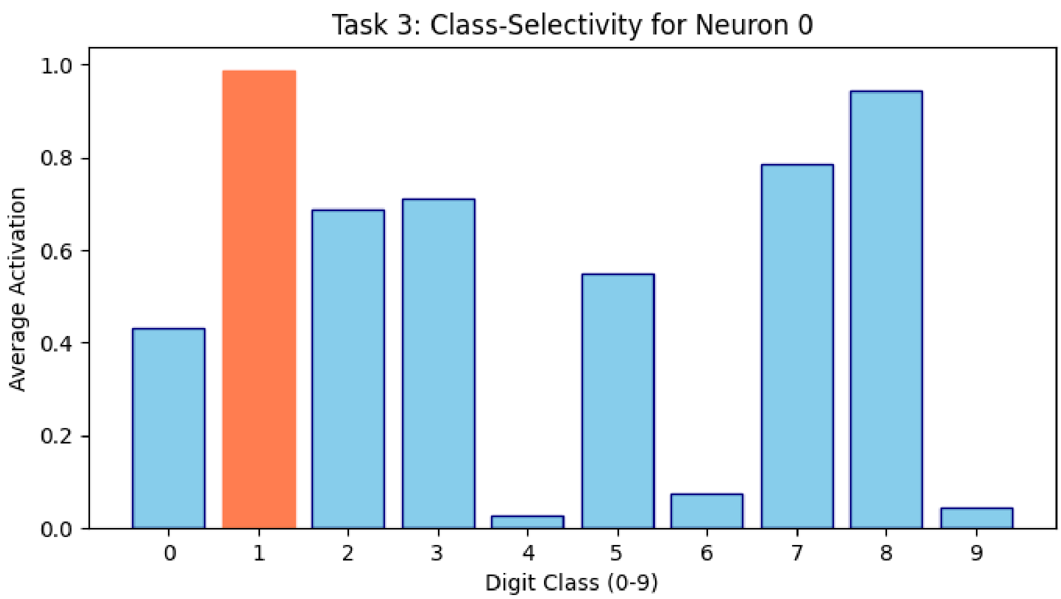
Informal Names for each Neuron :

Neuron 0	Upper Curve
Neuron 1	Curvy Diagonal
Neuron 2	Caught Three
Neuron 3	Bottom S
Neuron 4	Left Straight Stroke
Neuron 5	Saw C
Neuron 6	Confused
Neuron 7	Forward Slash
Neuron 8	Parallel Stroke
Neuron 9	Cap Wearer
Neuron 10	Snakey
Favourite Neuron 11	**Jalebi**
Neuron 12	Cross Lines
Neuron 13	Loop Detector
Neuron 14	Caught Five
Neuron 15	Diagonal Stroke
Neuron 16	Oblique line
Neuron 17	Half Circle
Neuron 18	Curvy Loop
Neuron 19	Caught Zero

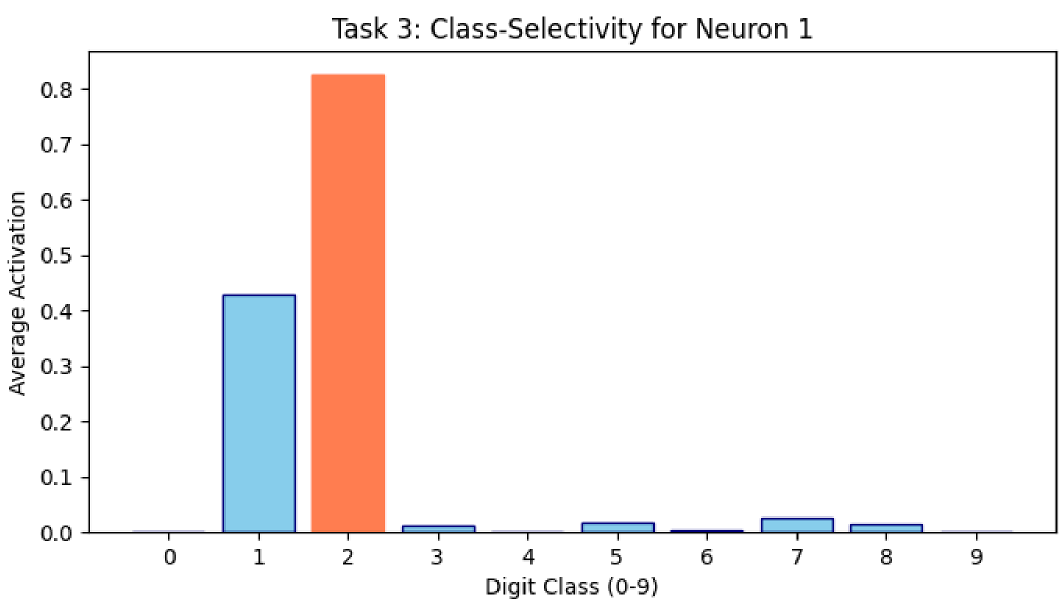
Conclusion:

The results from Task 2 demonstrate a clear correlation between the spatial weight patterns observed in Task 1 and the specific digit features that trigger the highest activations.

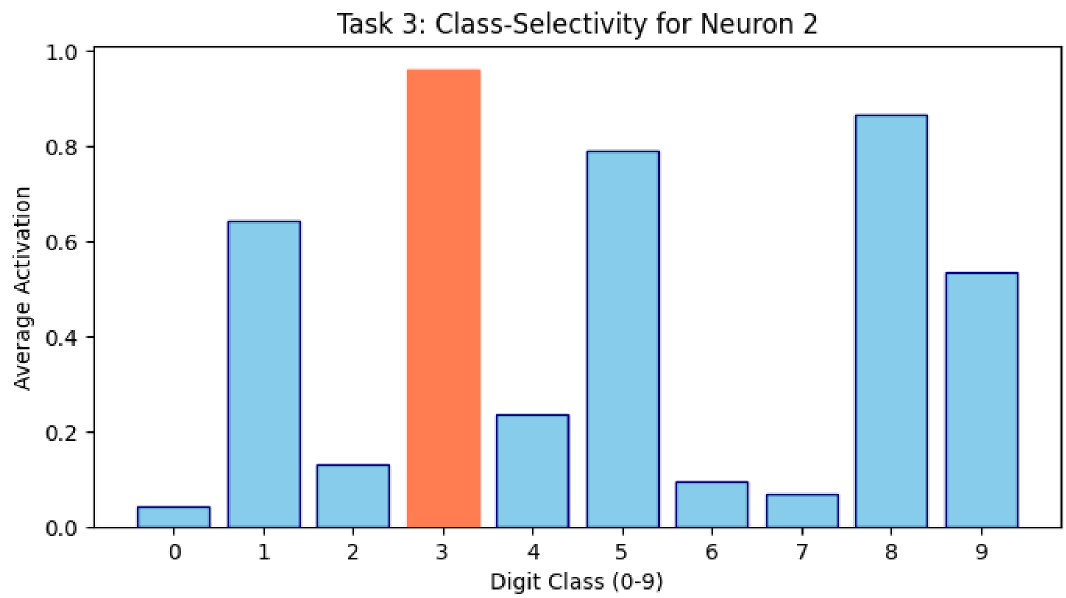
Task 3: Activation distribution and selectivity.



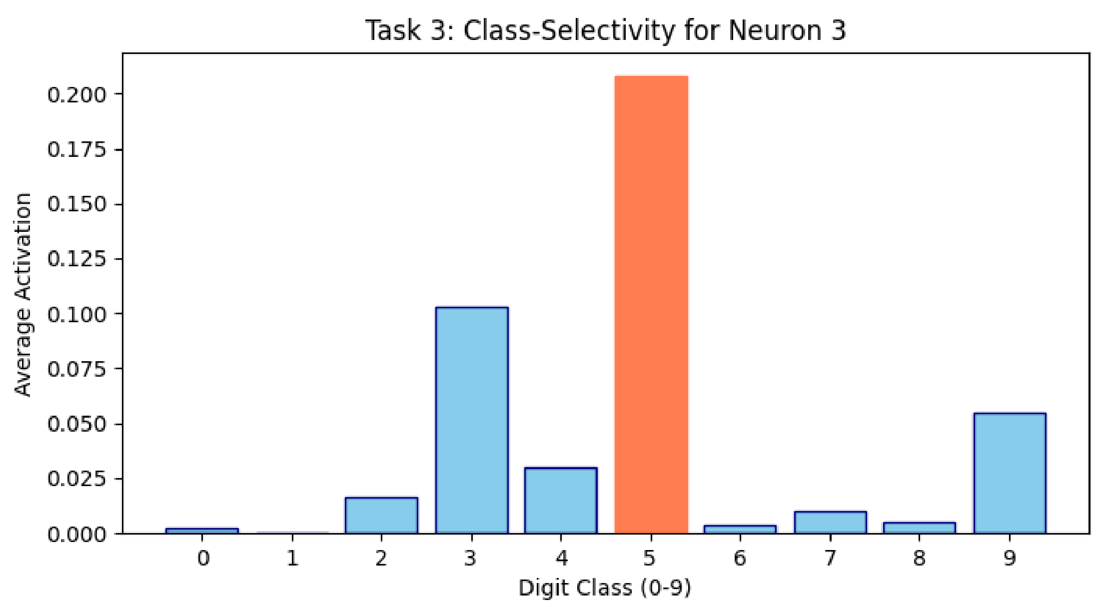
Neuron 0 is feature-selective because it acts as an "upper curved detector" for various digits like 0 and 2, rather than being exclusive to a single digit class 1.



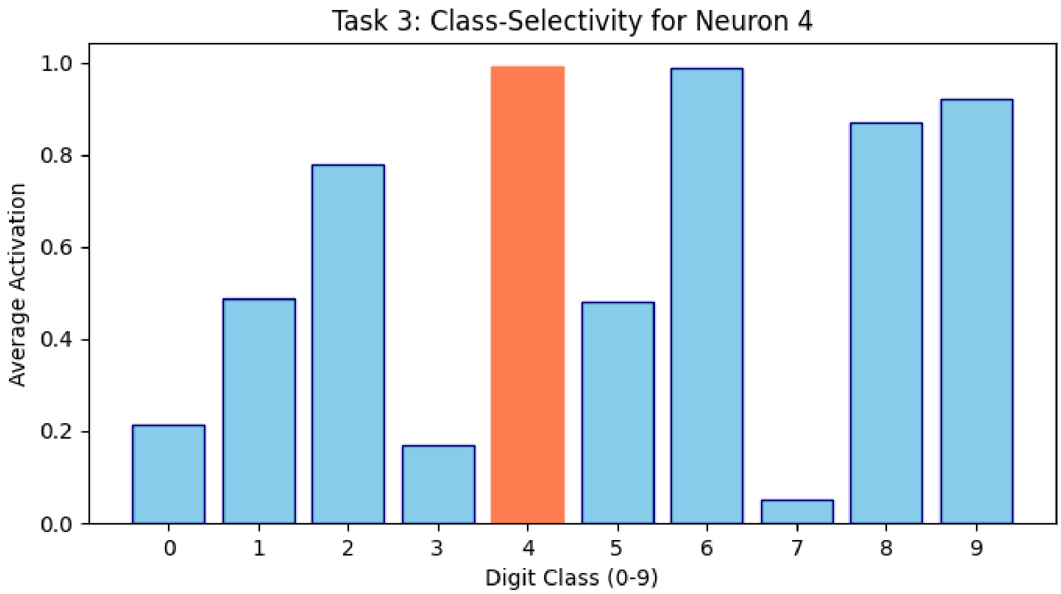
Neuron 1 is class-selective because it acts as a "Curvy Diagonal detector" showing a high average activation almost exclusively for class 2 and responding primarily to that label in its top inputs.



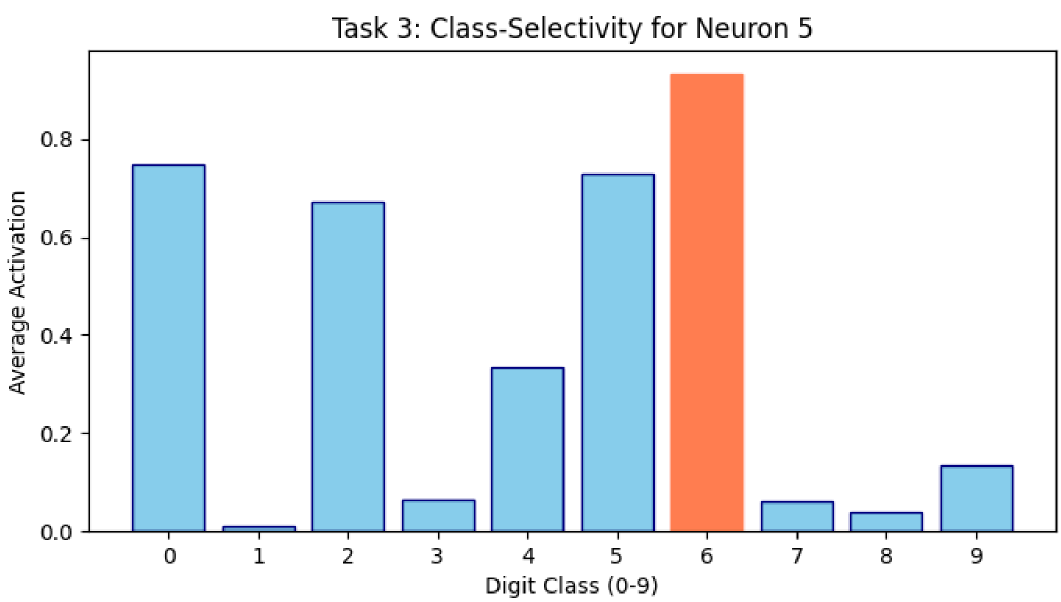
Neuron 2 is class-selective because it behaves primarily as a "digit 3 detector" demonstrating nearly exclusive average activation for class 3 and responding only to that label in its top inputs



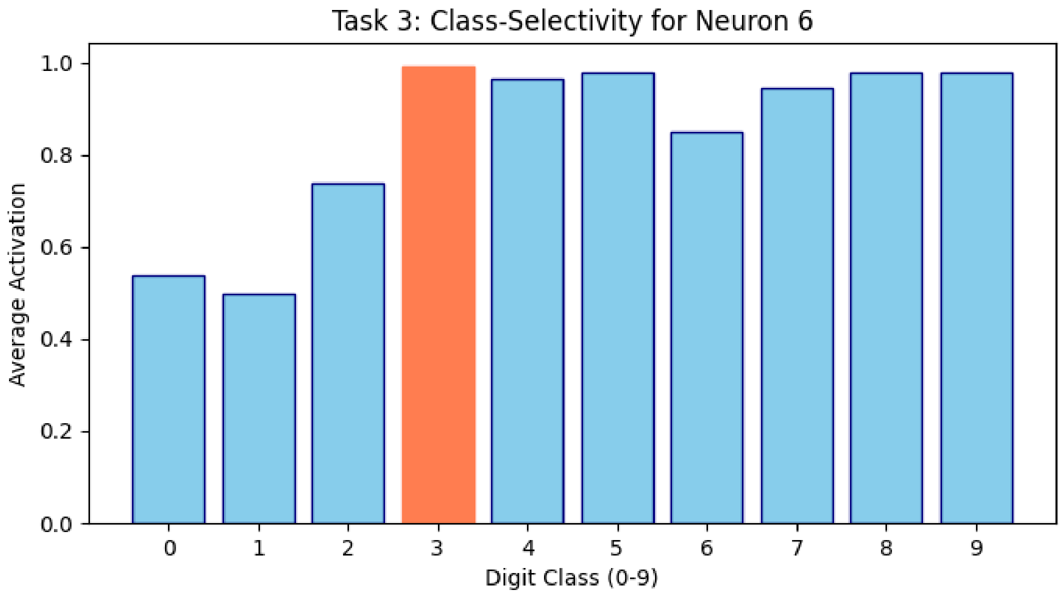
Neuron 3 is feature-selective as it identifies the top horizontal bar shared by digits 3 and 5, resulting in mixed top-activating images despite a slightly higher average activation for class 5.



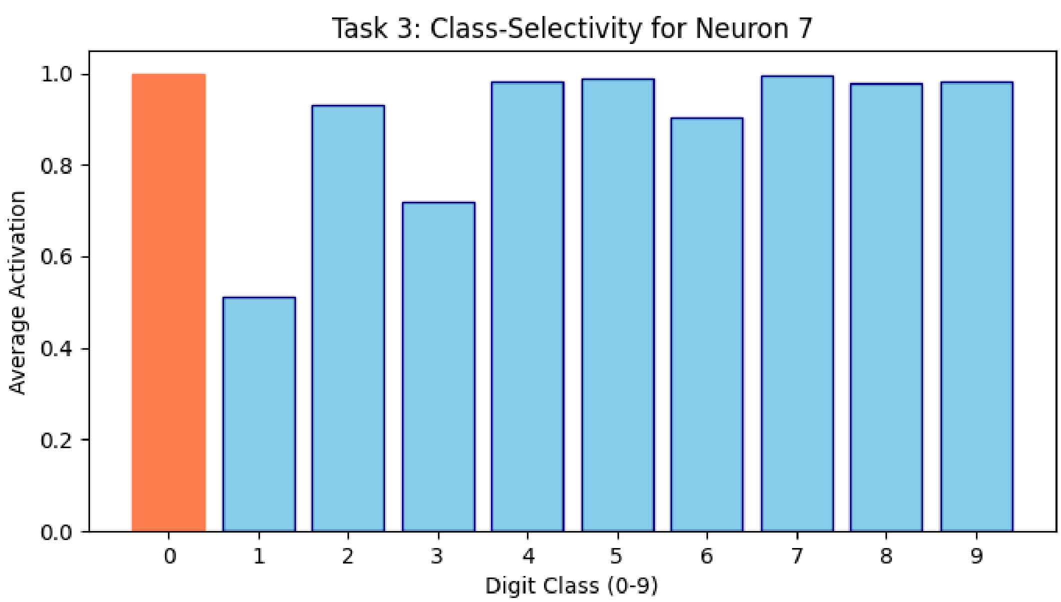
Neuron 4 is feature-selective because it acts as a "left-straight stroke" shared by digits 4 and 6, resulting in mixed top-activating images despite a high average activation for multiple classes.



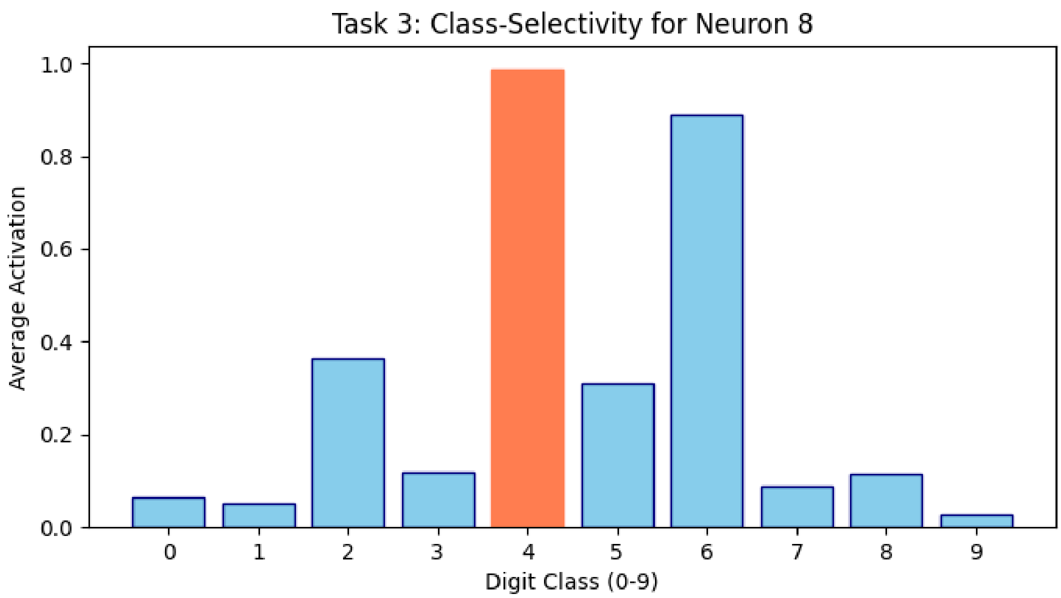
Neuron 5 is class-selective because it acts as a "digit 6 detector," showing a clear preference for class 6 in its average activation and responding almost exclusively to that label in its top inputs.



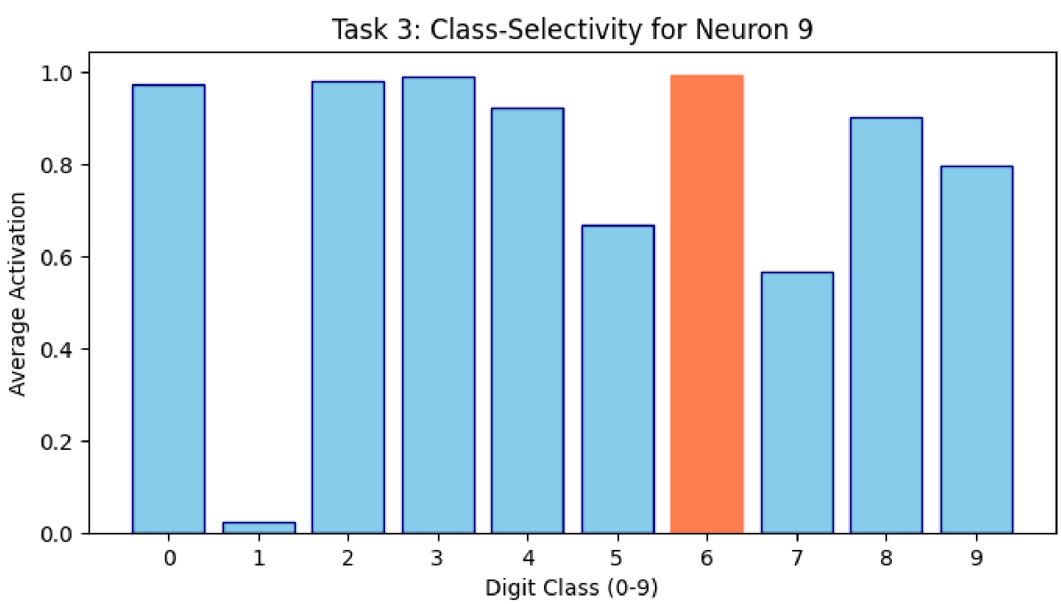
Neuron 6 is feature-selective as it responds to broad diagonal strokes and edges across a wide variety of digit classes including 5, 4, 3, 0, 6, and 7.



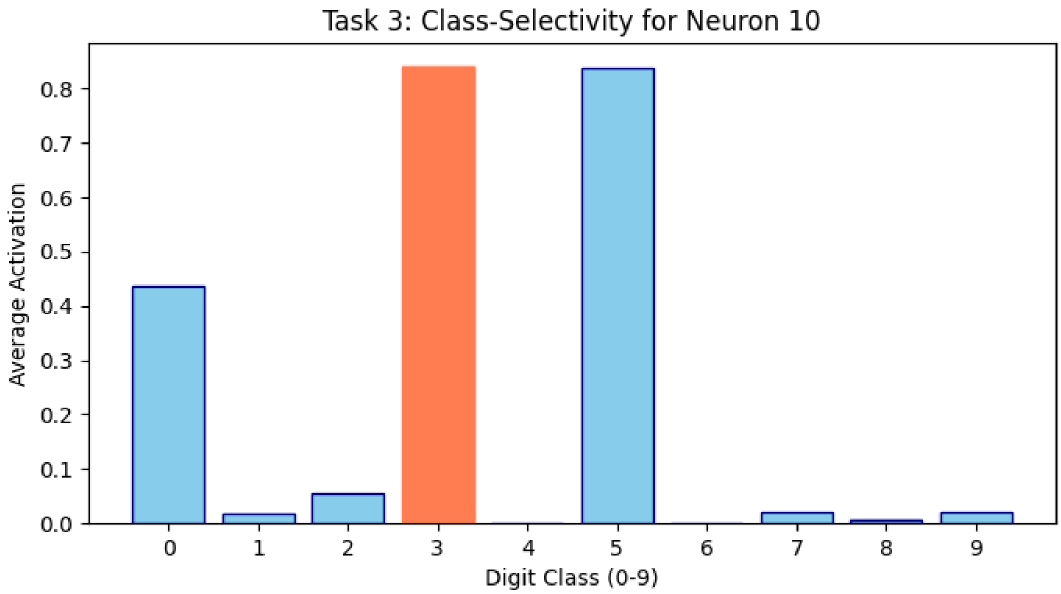
Neuron 7 is feature-selective because it triggers on sharp diagonal strokes and upright lines found across many digit classes, including 4, 1, 9, 2, 3, and 7.



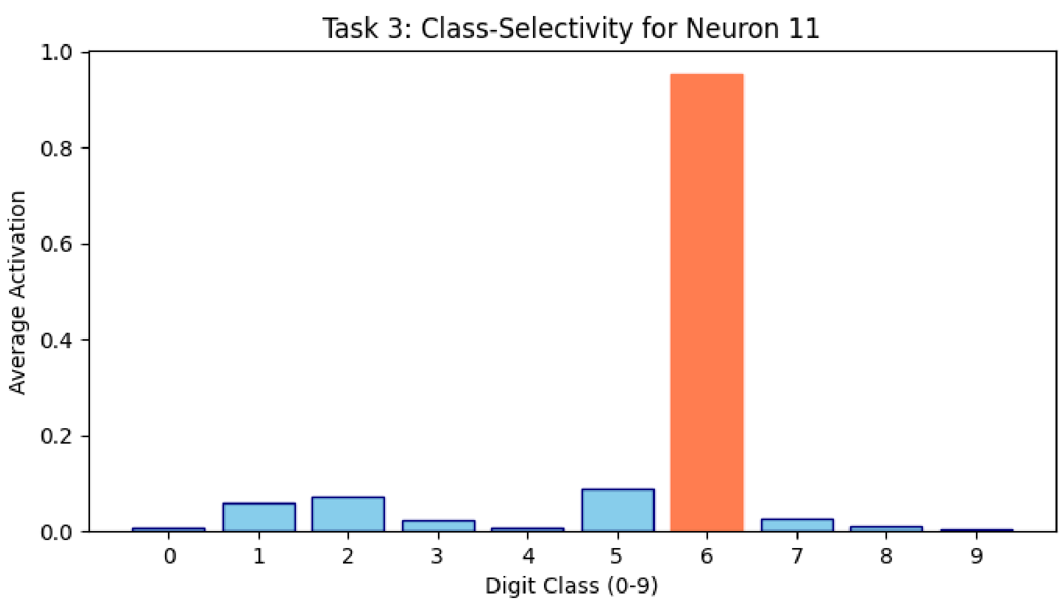
Neuron 8 is class-selective because it acts as a "digit 4 detector," showing its highest average activation for class 4 and responding almost exclusively to that label in its top inputs.



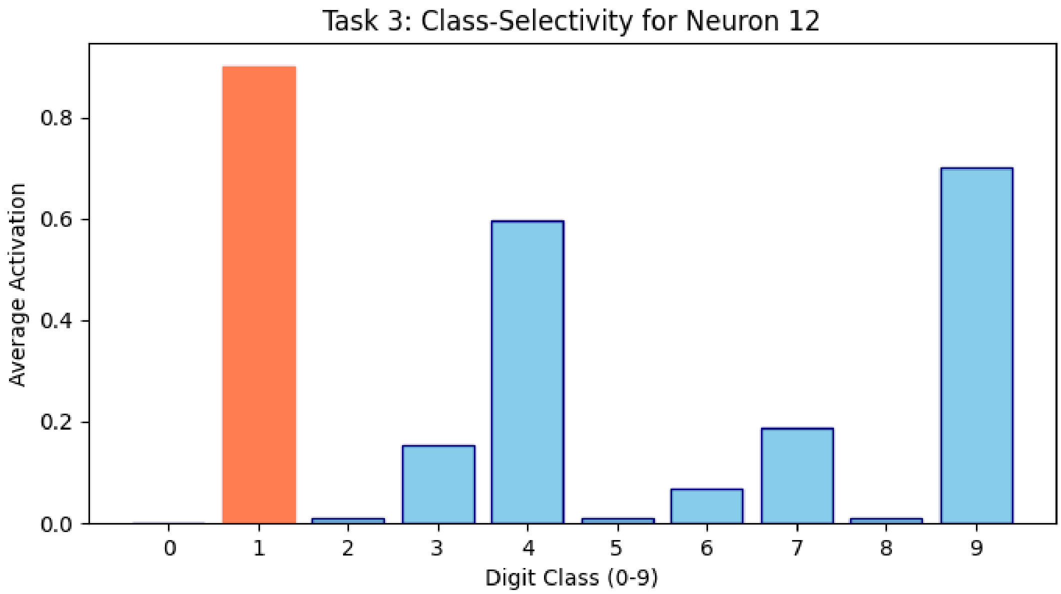
Neuron 9 is feature-selective as it responds to shared curved primitives and horizontal strokes found across multiple digit classes, including 0, 2, 3, 4, 8, and 9.



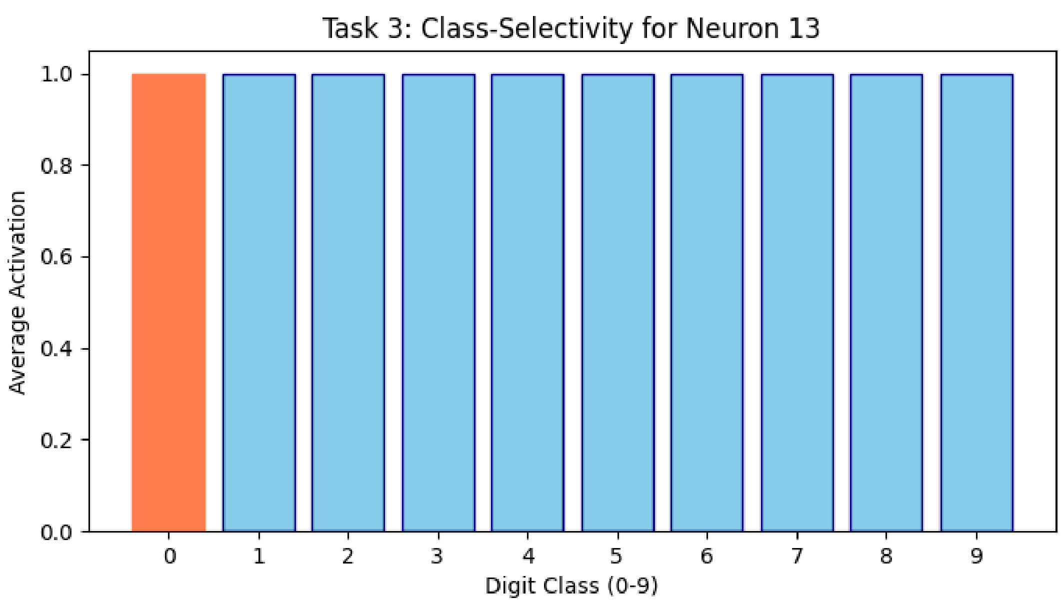
Neuron 10 is feature-selective as it responds to concave curves and horizontal middle strokes shared by digits 3 and 5.



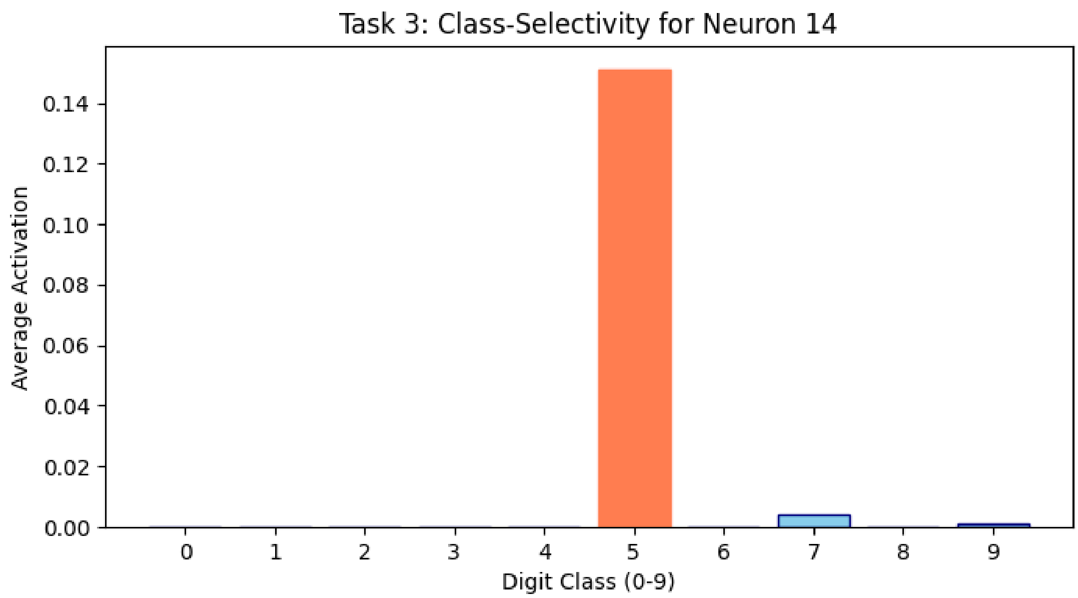
Favourite Neuron 11 is class-selective because it acts as a "digit 6 detector," showing its highest average activation almost exclusively for class 6 and responding primarily to that label in its top inputs.



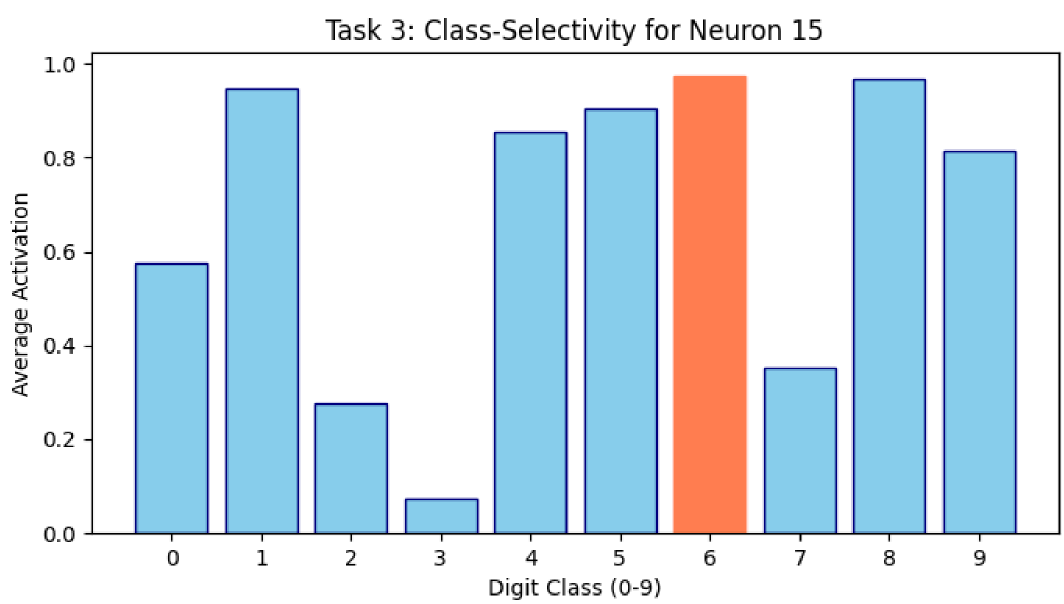
Neuron 12 is feature-selective as it responds to upright vertical strokes and sharp angles common across multiple digit classes, including 4, 7, 2, and 9.



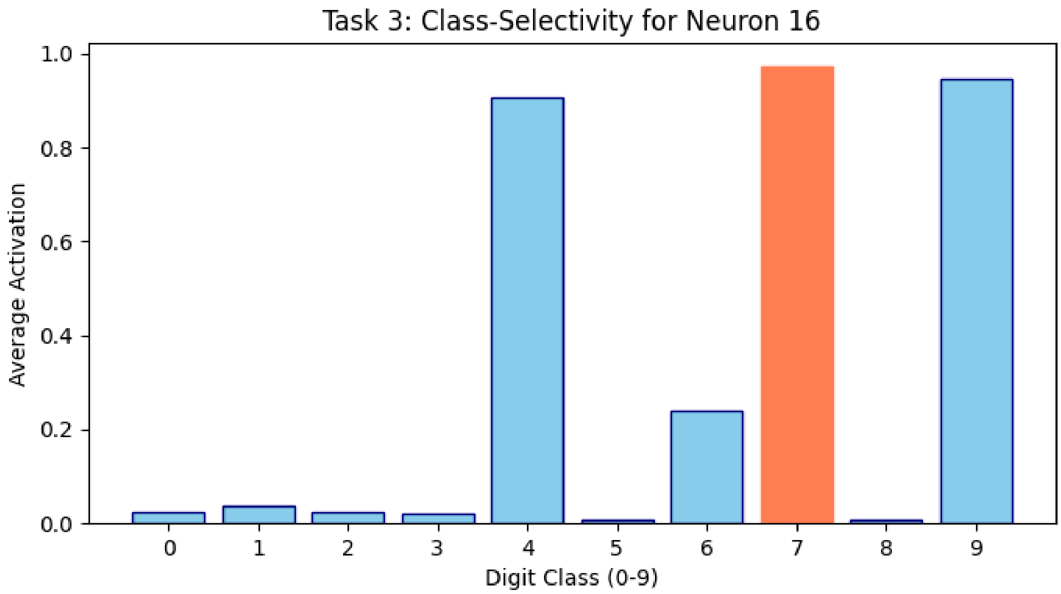
Neuron 13 is feature-selective as it shows near-identical average activation across all digit classes, acting as a highly generalized background or edge detector.



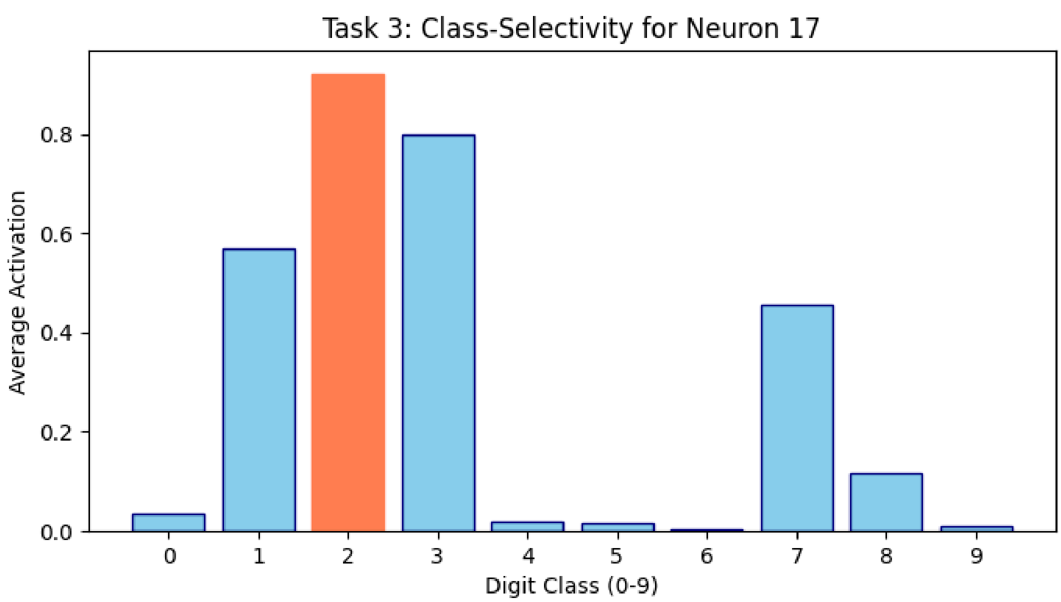
Neuron 14 is class-selective because it acts as a "digit 5 detector," showing its highest average activation almost exclusively for class 5 and responding only to that label in its top inputs.



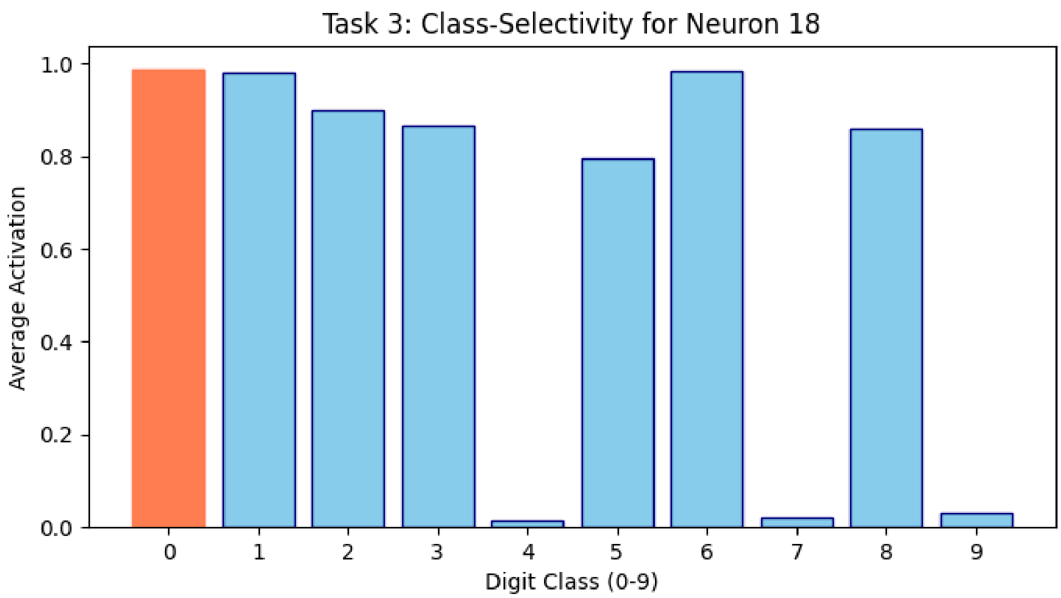
Neuron 15 is feature-selective as it responds to sharp vertical strokes and loops common to digits 1, 4, 6, 8, and 9.



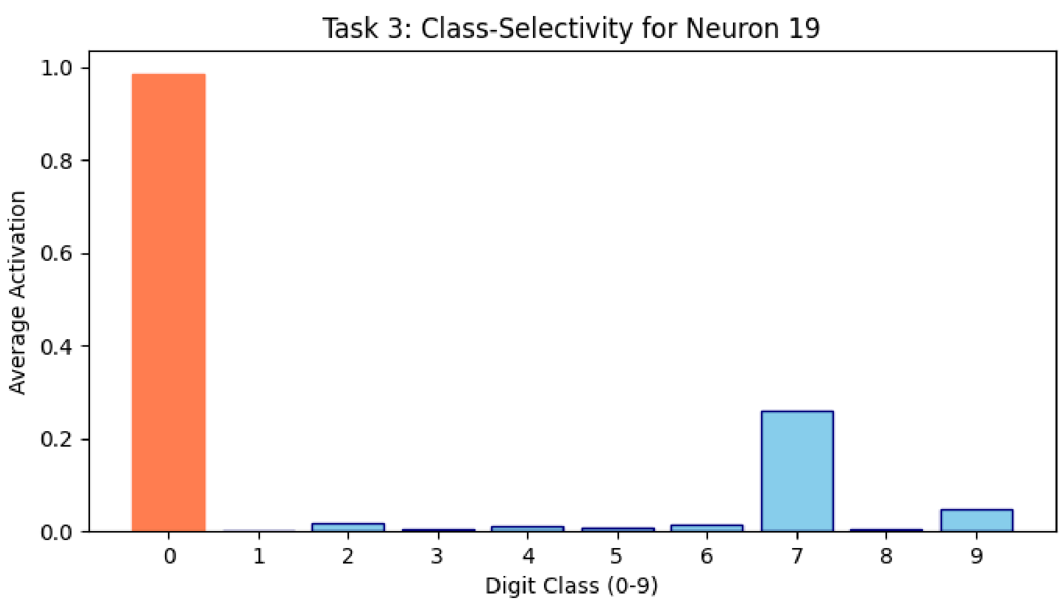
Neuron 16 is feature-selective as it identifies the top horizontal bar and sharp angles shared by digits 4, 7, and 9.



Neuron 17 is feature-selective as it triggers on horizontal curved strokes found in both digits 2 and 3.



Neuron 18 is feature-selective as it responds to generalized curves and rounded components shared across diverse digit classes including 0, 1, 2, 3, 5, 6, and 8.



Neuron 19 is class-selective because it acts as a "digit 0 detector," showing a high average activation almost exclusively for class 0 and responding only to that label in its top inputs.