Assignment 4

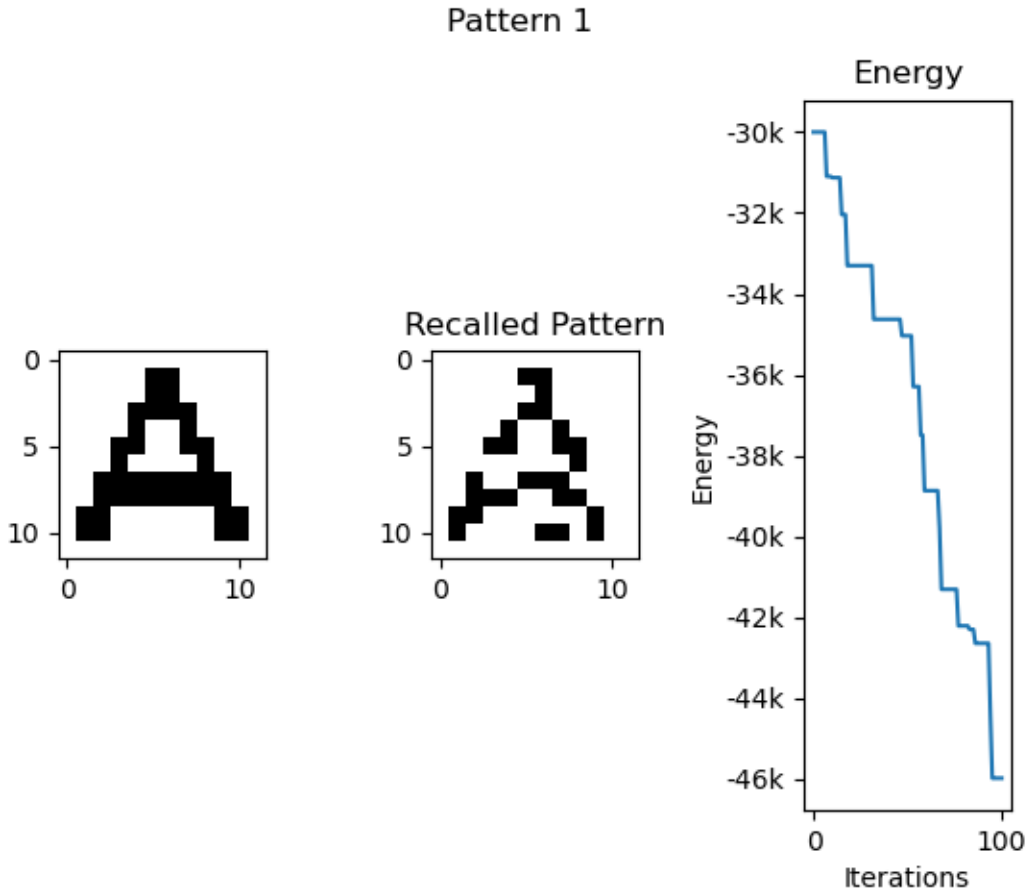
Advanced methods in machine learning

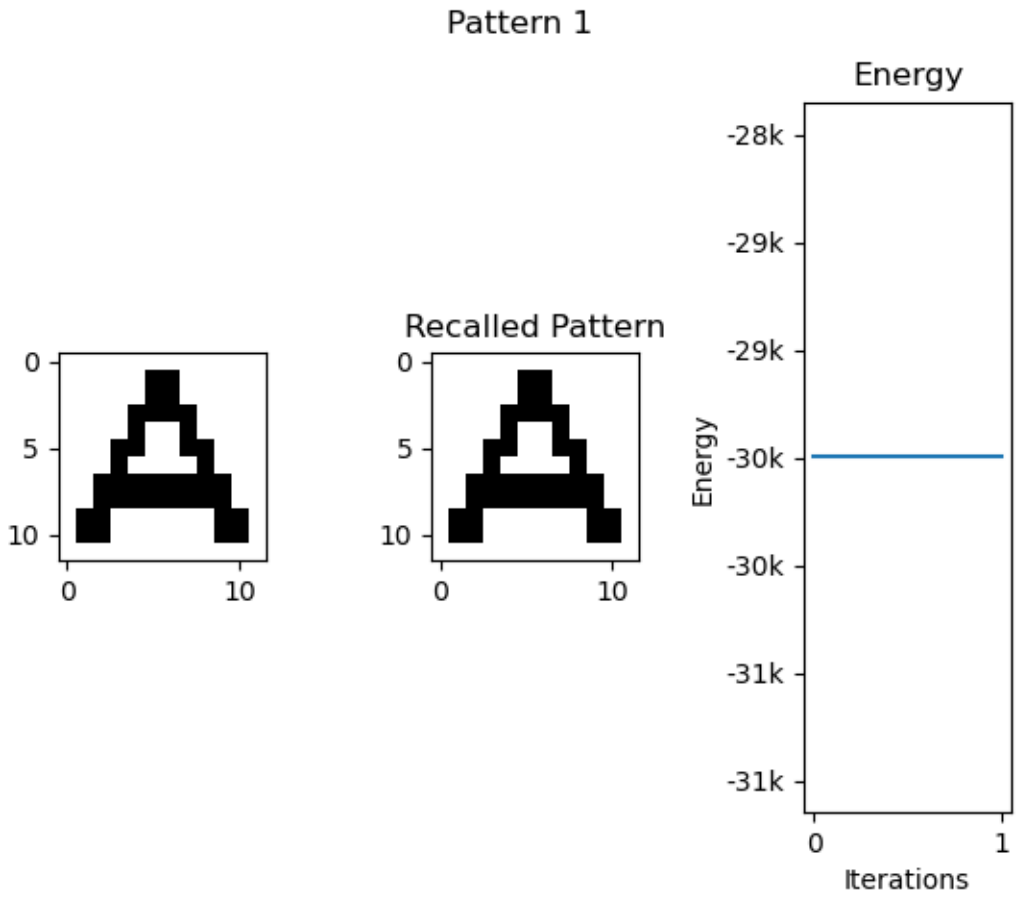
Hopfield network

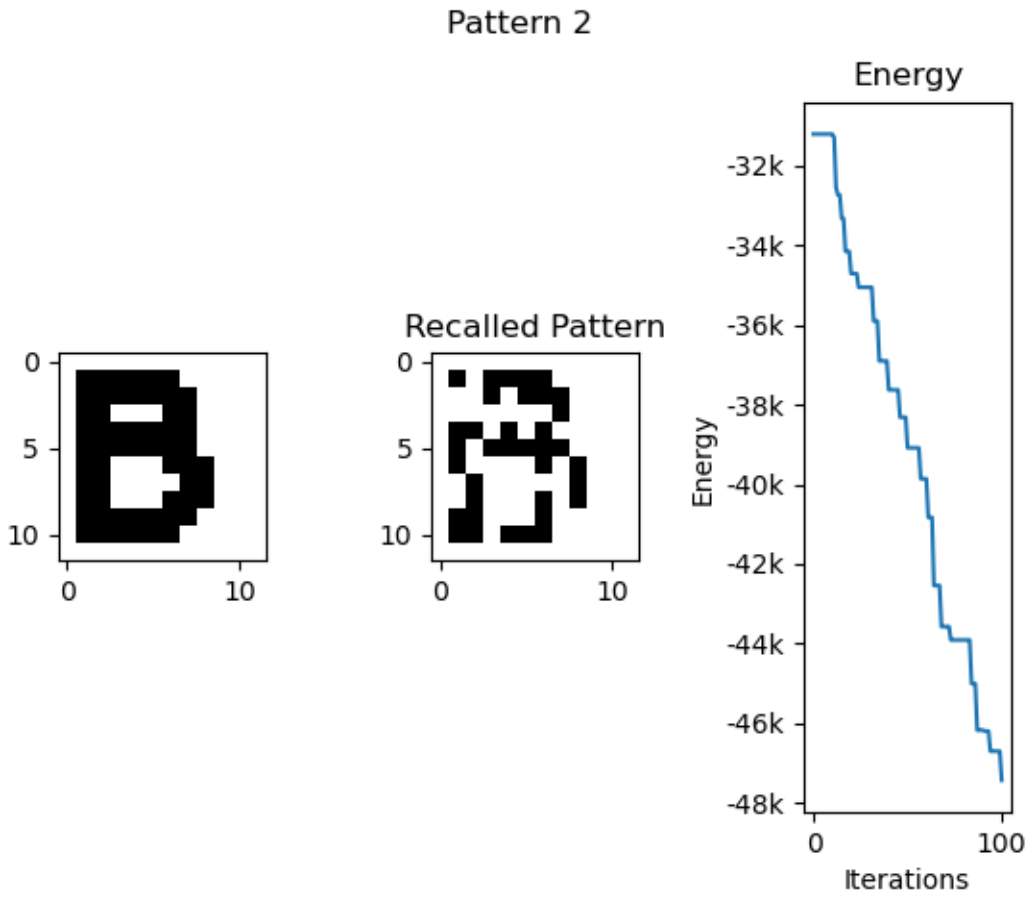
Throughout the course of working on this project we came across many issues with pattern recognition both on original (training) and test patterns. Many interpretations of the update rule were attempted (synchronous, asynchronous, with and without early stopping, early stopping with multiple stable “updates” etc.) the interpretation attached is synchronous with early stopping.

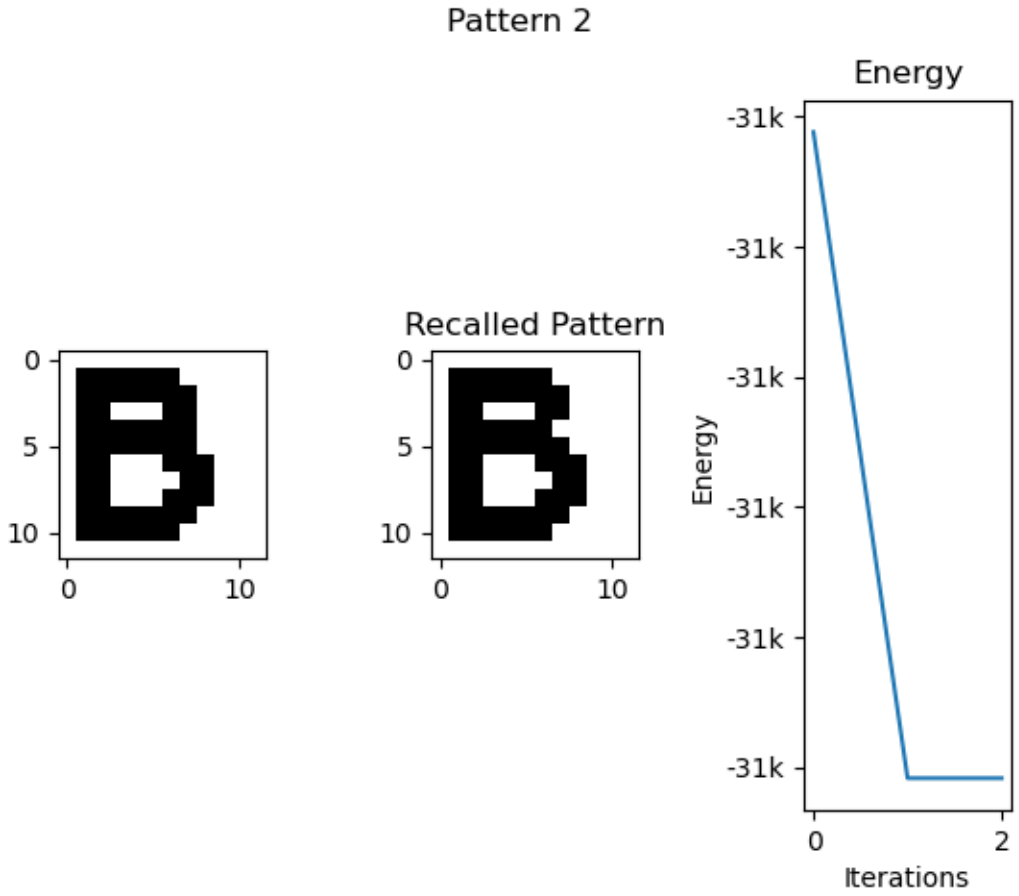
as you will see when using early stopping, the network stops updating in one or 2 itterations and spits out a image that is nearly identical to the querry, wether it is a training pattern or one of the test patterns with 10% noise. But without early stopping, you see that the energy never really converges and after 100 updates the image looks nothing like one of the training patterns. Our theory is that the data being trained on is to similar to one another and that we have more patterns than the network with this many neurons can handle.

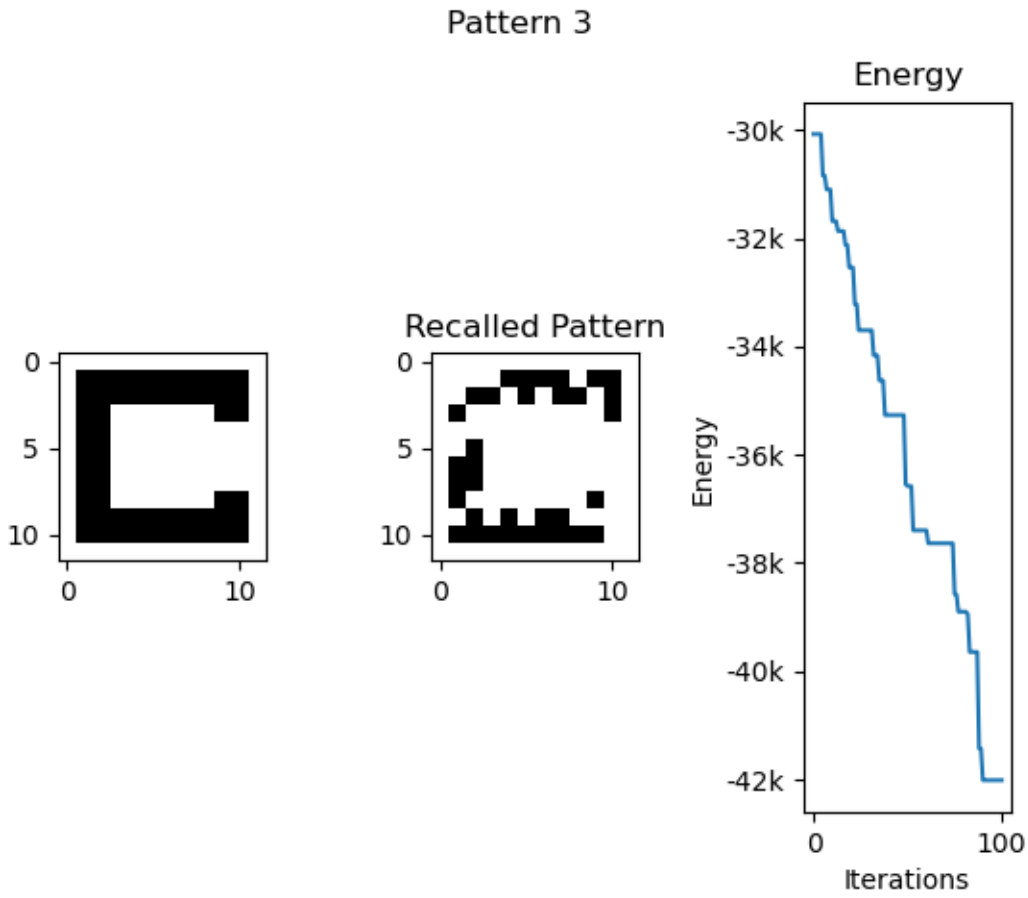
Here are the images of recall and energy on input images, both with and without early stopping.

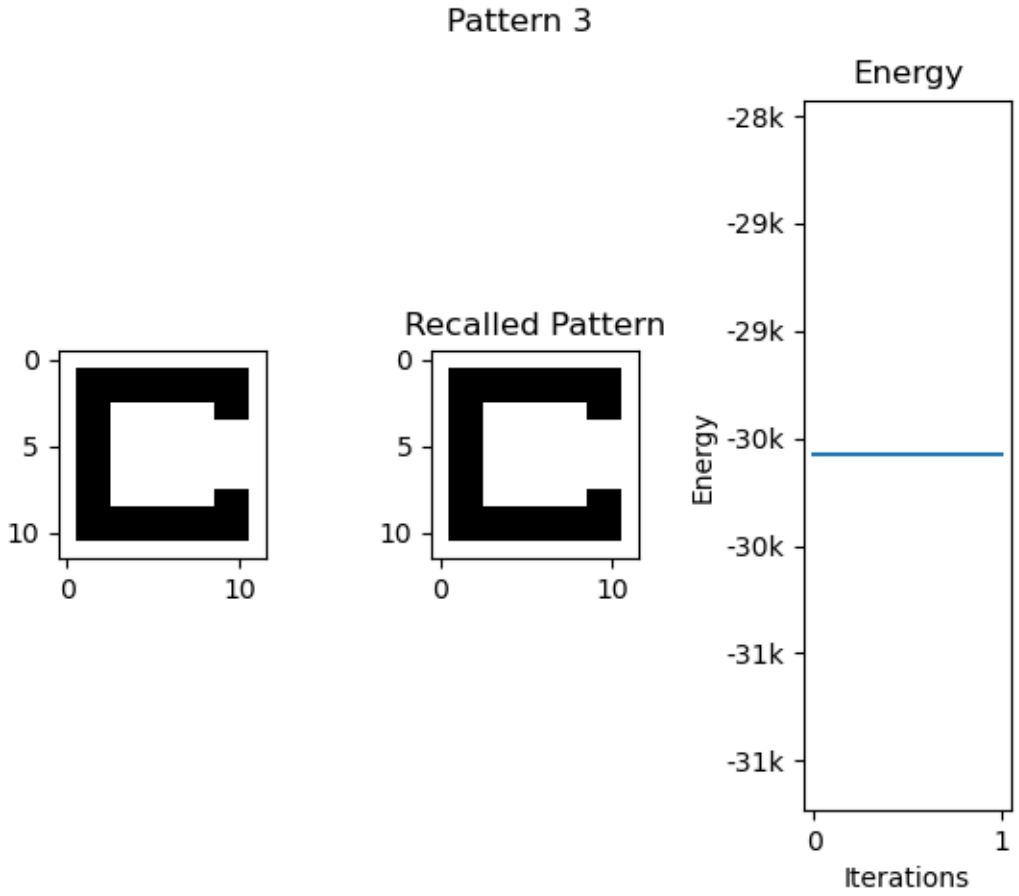

Pattern 1
Recalled Pattern
Energy
0
5
10
0
10
-30k
-32k
-34k
-36k
-38k
-40k
-42k
-44k
-46k
Energy
0
100
Iterations

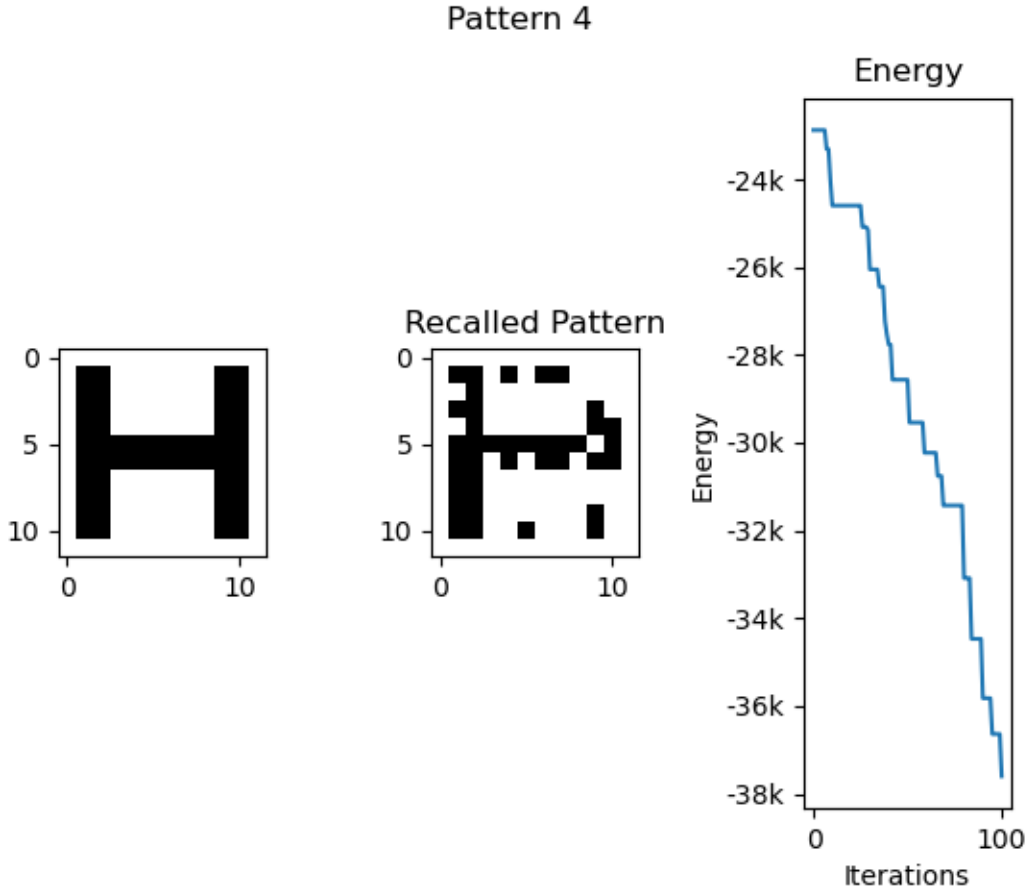

Pattern 1
Recalled Pattern
Energy
0
5
10
0
10
-28k
-29k
-29k
-30k
-30k
-31k
-31k
Energy
0
1
Iterations

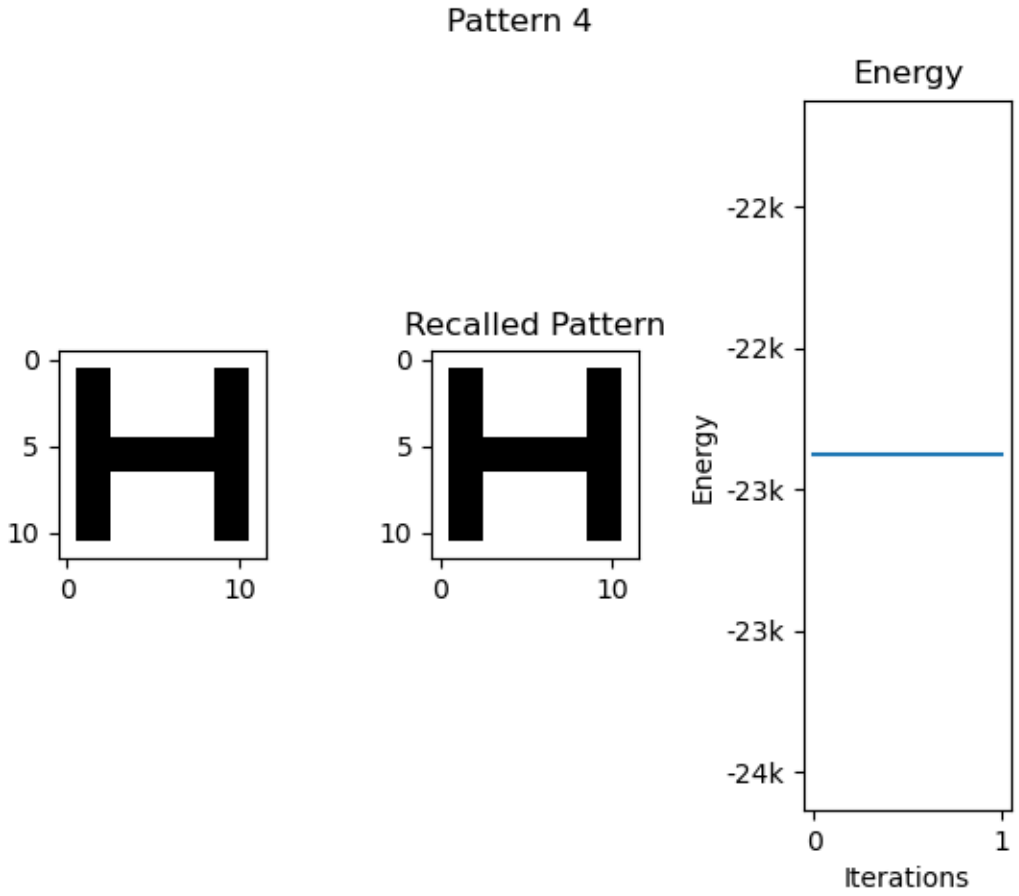

Pattern 2
Recalled Pattern
Energy
0
5
10
0
10
-32k
-34k
-36k
-38k
-40k
-42k
-44k
-46k
-48k
Energy
0
100
Iterations

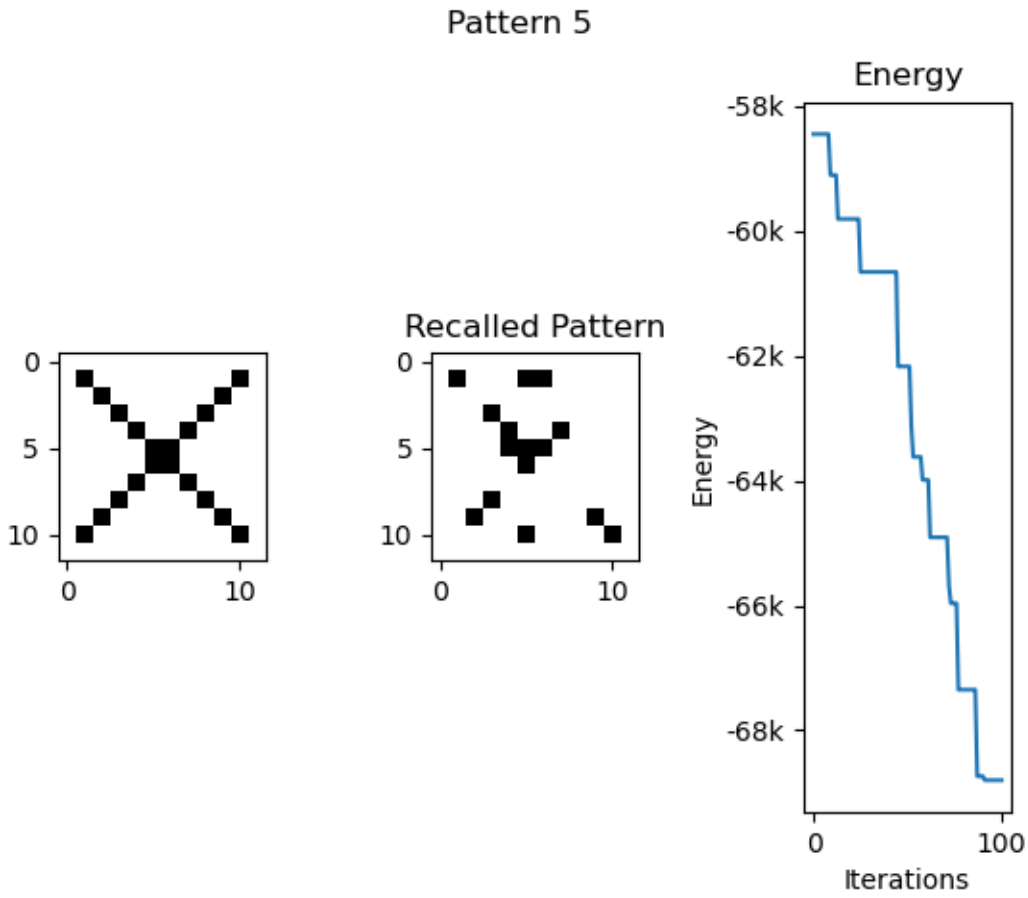

Pattern 2
Recalled Pattern
Energy
0
5
10
0
10
-31k
-31k
-31k
-31k
-31k
Energy
0
2
Iterations

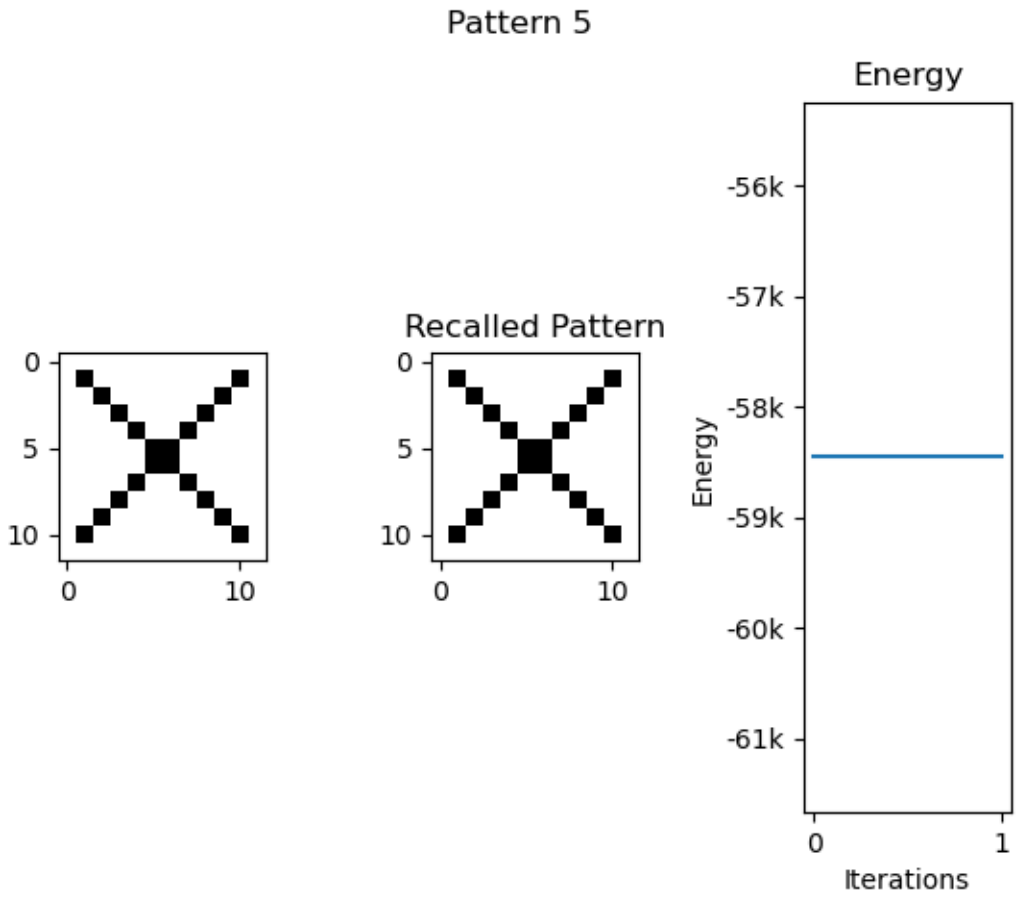

Pattern 3
Recalled Pattern
Energy
0
5
10
0
10
-30k
-32k
-34k
-36k
-38k
-40k
-42k
Energy
0
100
Iterations

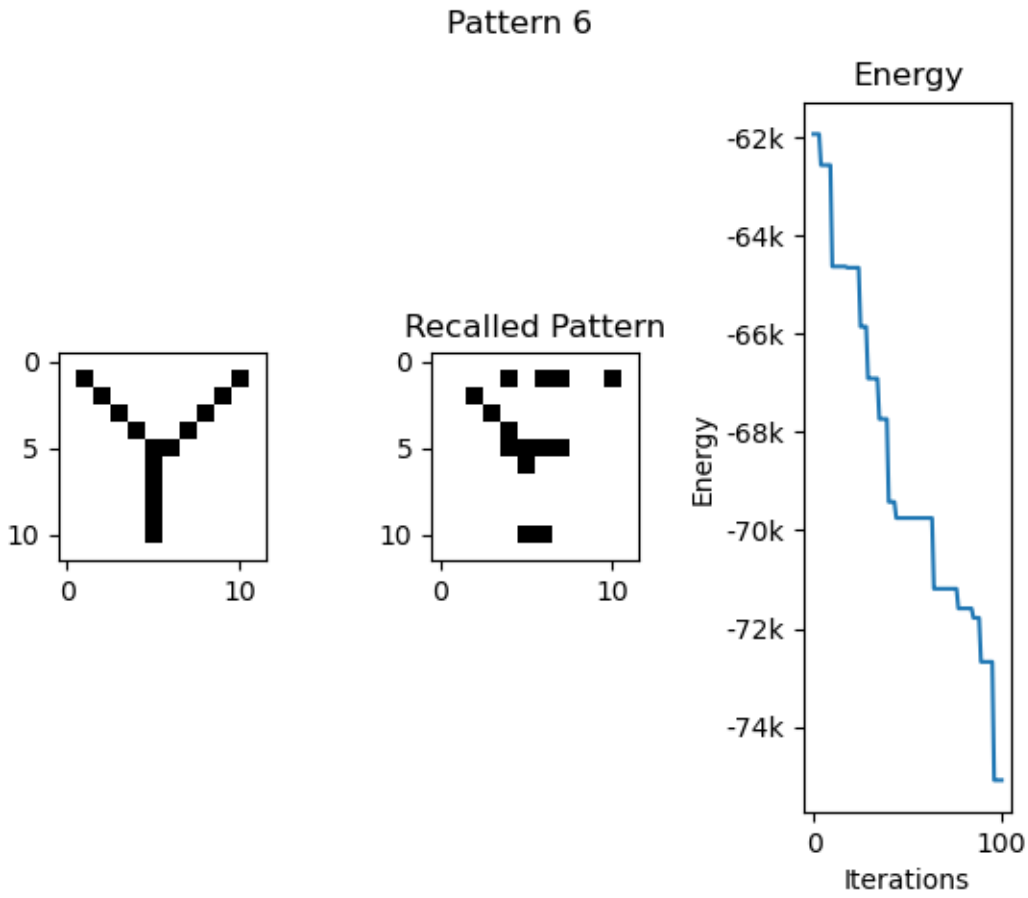

Pattern 3
Recalled Pattern
Energy
0
5
10
0
10
-28k
-29k
-29k
-30k
-30k
-31k
-31k
Energy
0
1
Iterations

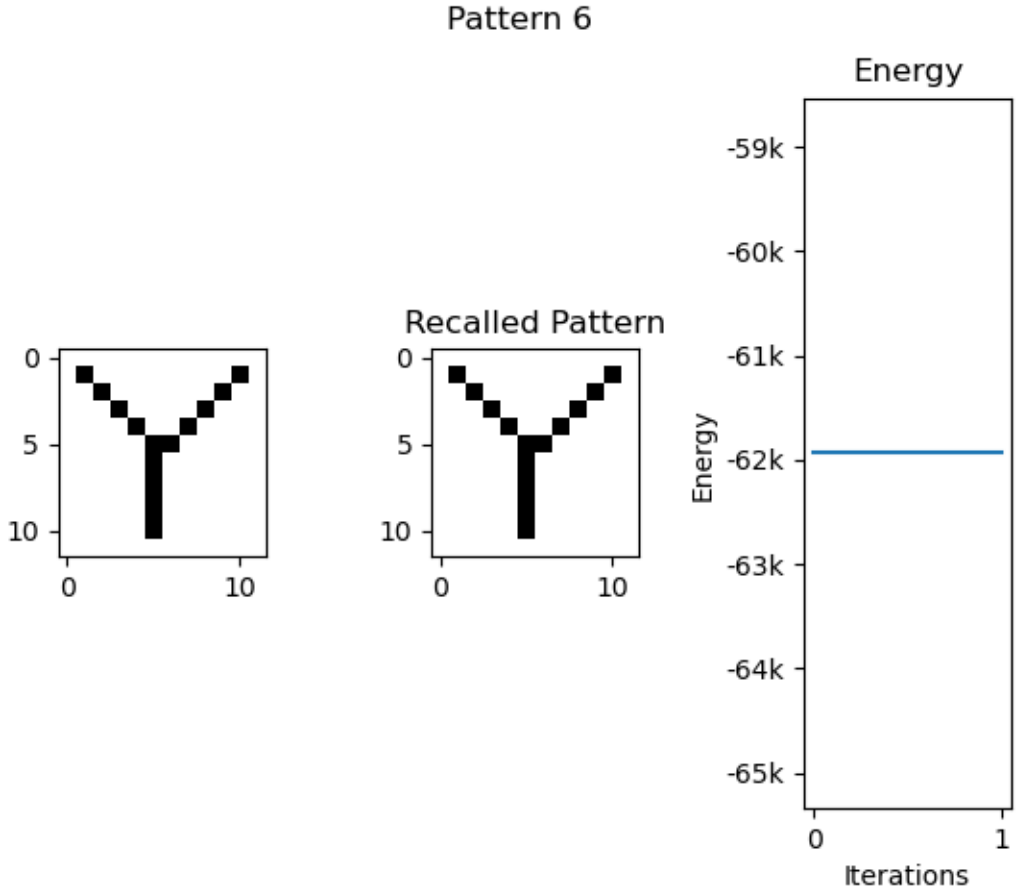

Pattern 4
Recalled Pattern
Energy
0
5
10
0
10
-24k
-26k
-28k
-30k
-32k
-34k
-36k
-38k
Energy
0
100
Iterations

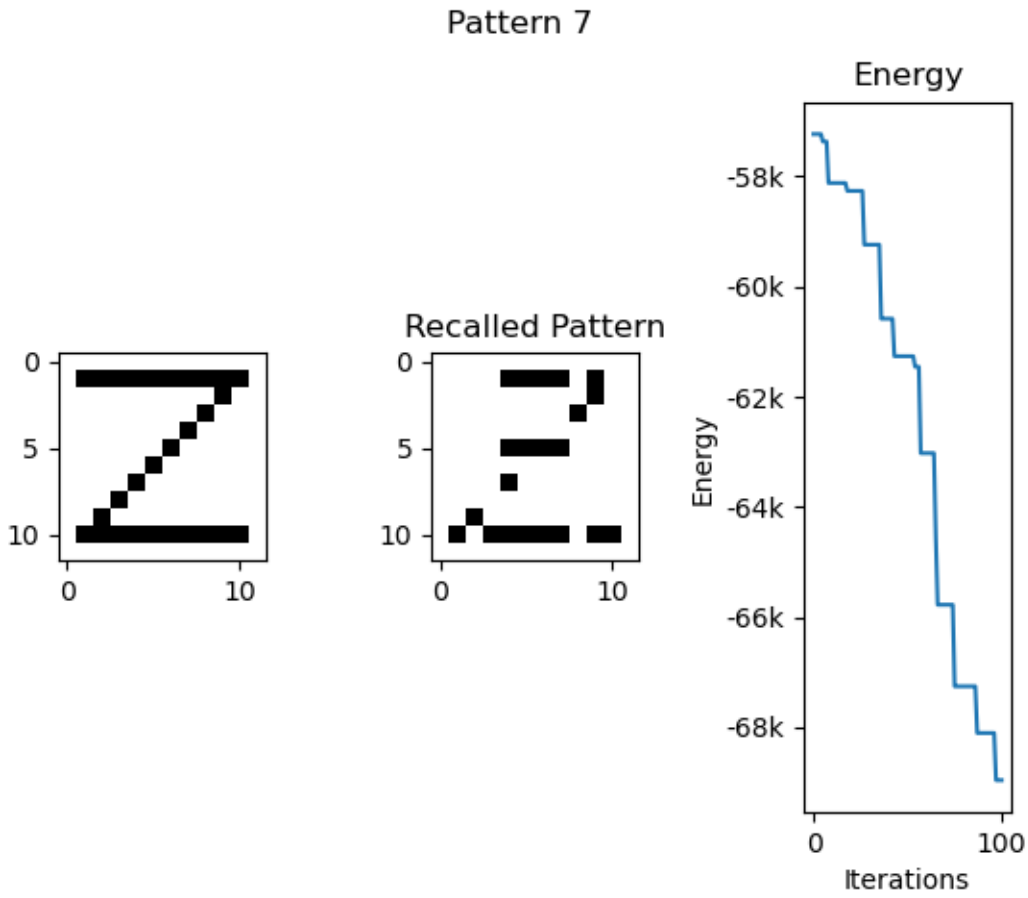

Pattern 4
Recalled Pattern
Energy
0
5
10
0
10
-22k
-22k
-23k
-23k
-24k
Energy
0
1
Iterations

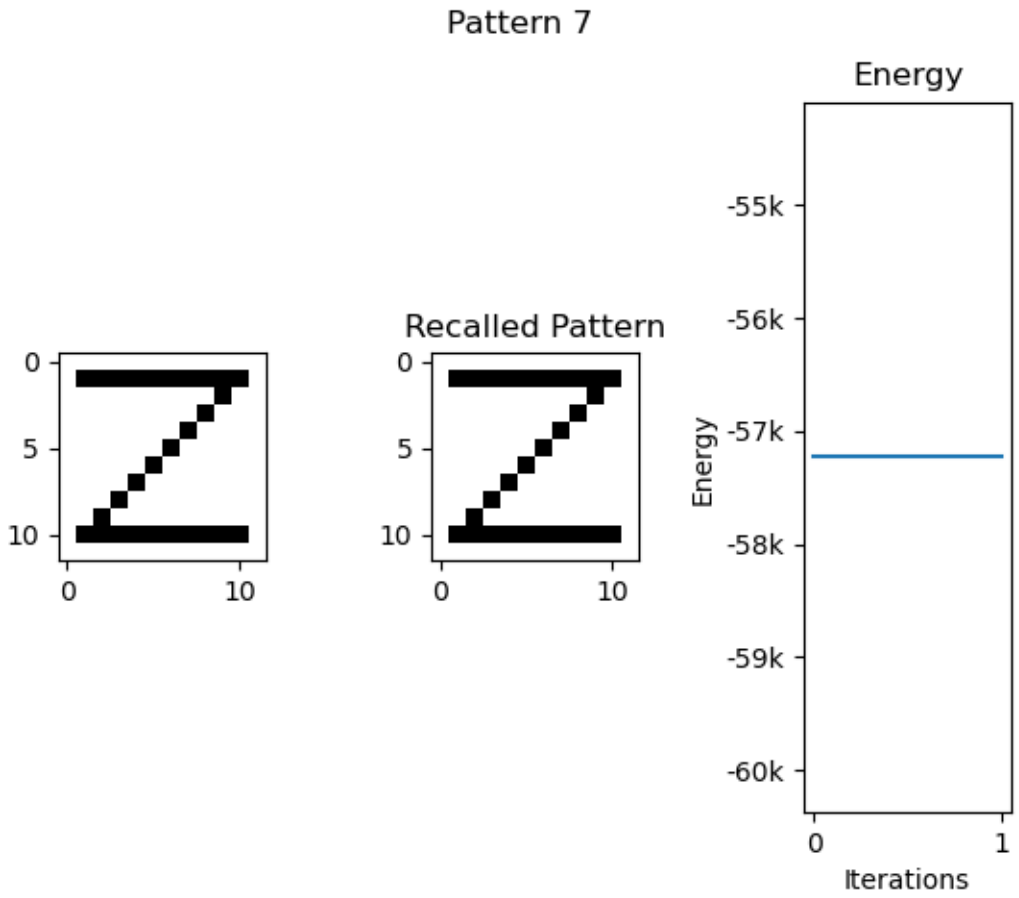

Pattern 5
Recalled Pattern
Energy
0
5
10
0
10
-58k
-60k
-62k
-64k
-66k
-68k
Energy
0
100
Iterations

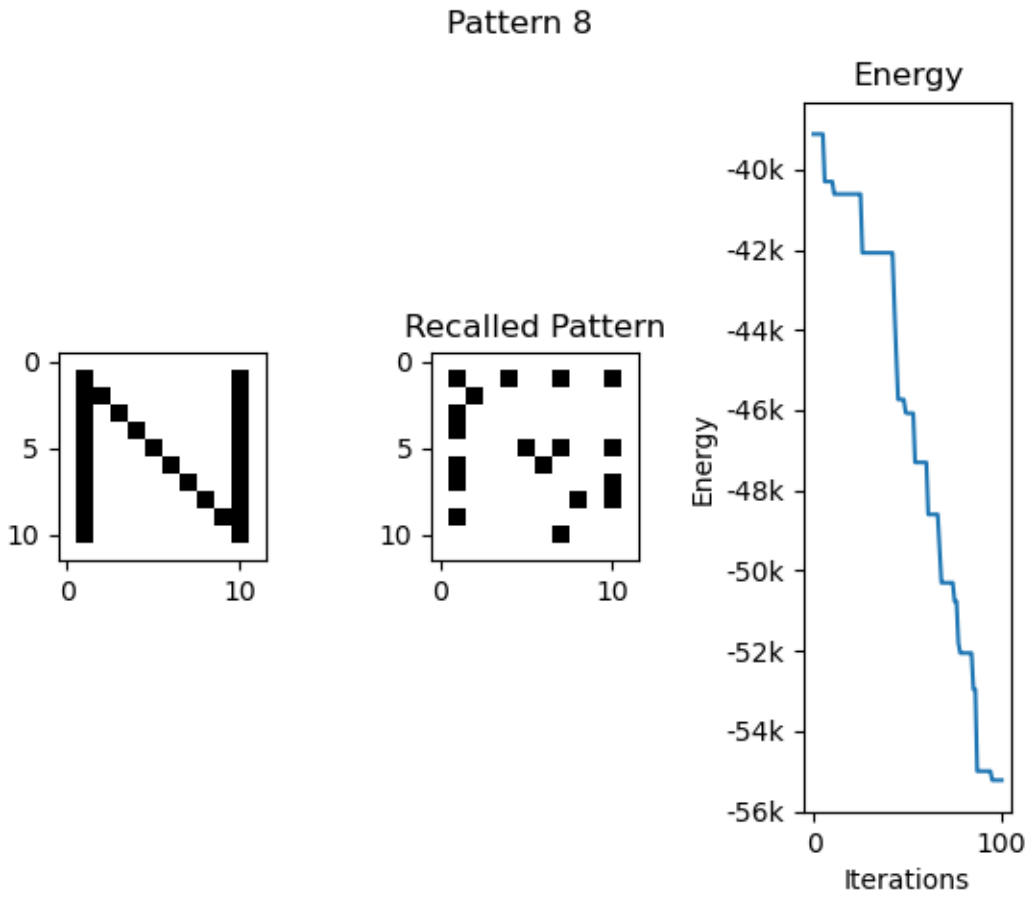

Pattern 5
Recalled Pattern
Energy
0
5
10
0
10
-56k
-57k
-58k
-59k
-60k
-61k
Energy
0
1
Iterations

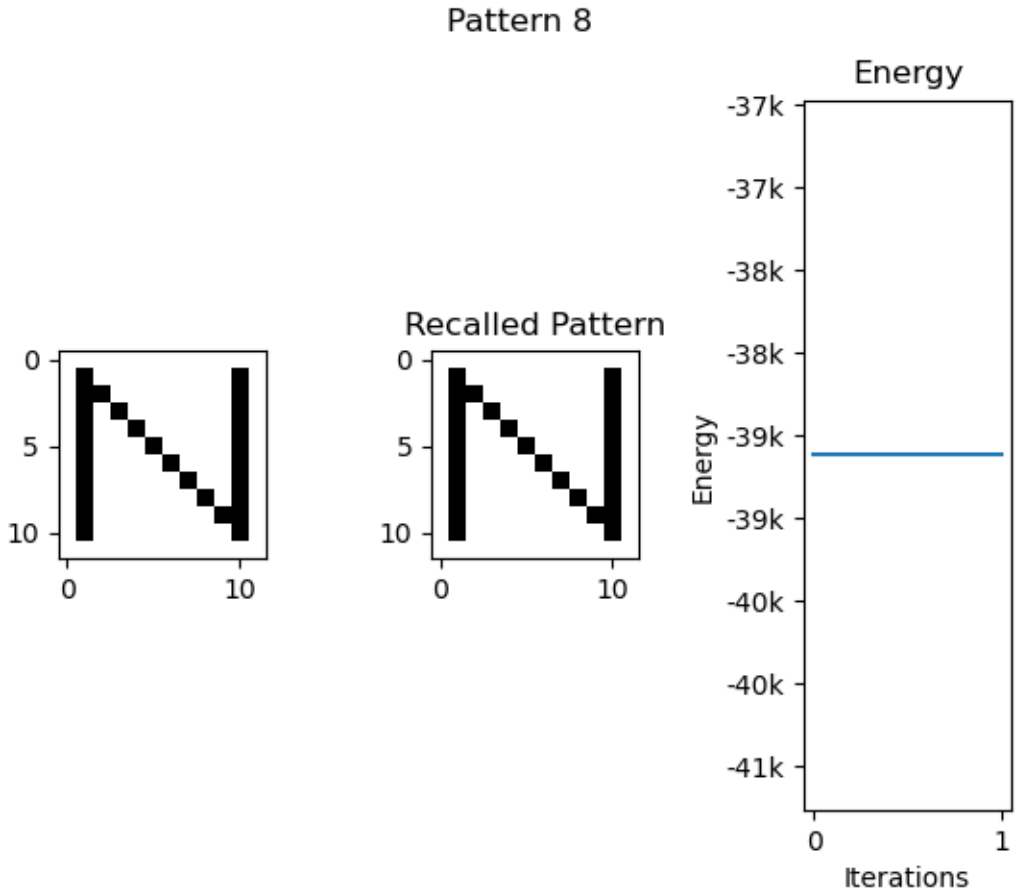

Pattern 6
Recalled Pattern
Energy
0
5
10
0
10
-62k
-64k
-66k
-68k
-70k
-72k
-74k
Energy
0
100
Iterations

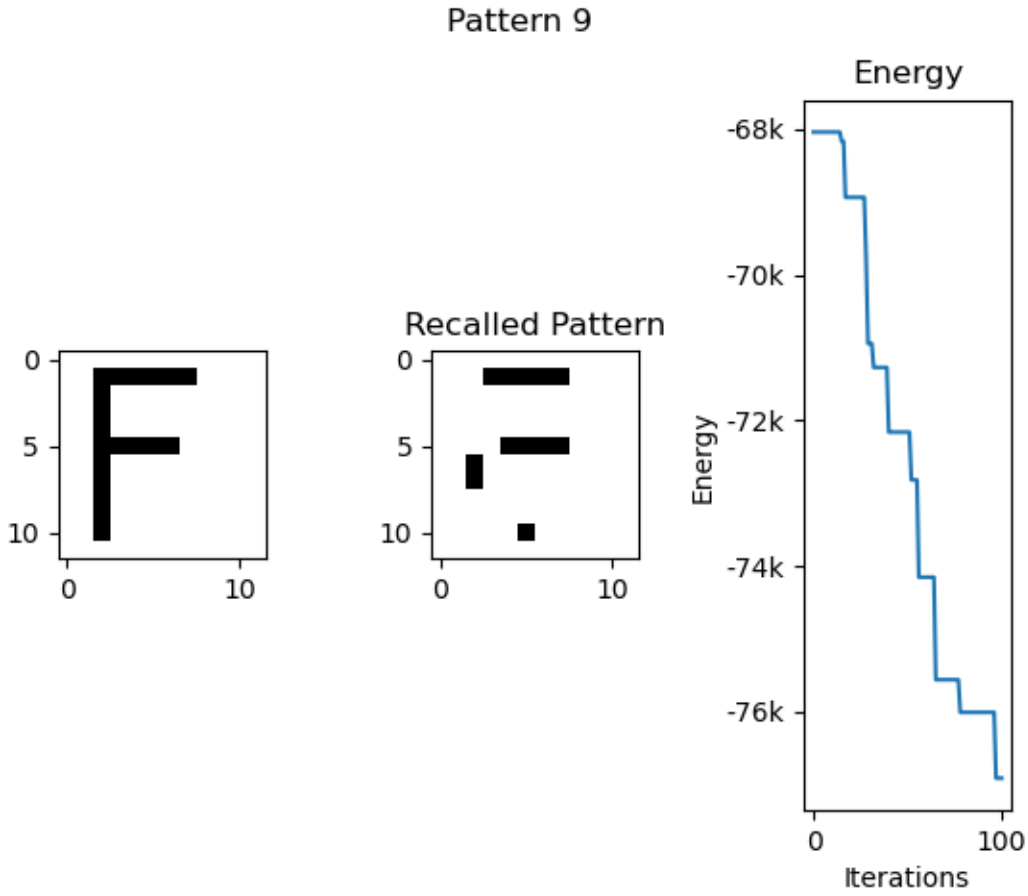

Pattern 6
Recalled Pattern
Energy
0
5
10
0
10
-59k
-60k
-61k
-62k
-63k
-64k
-65k
Energy
0
1
Iterations

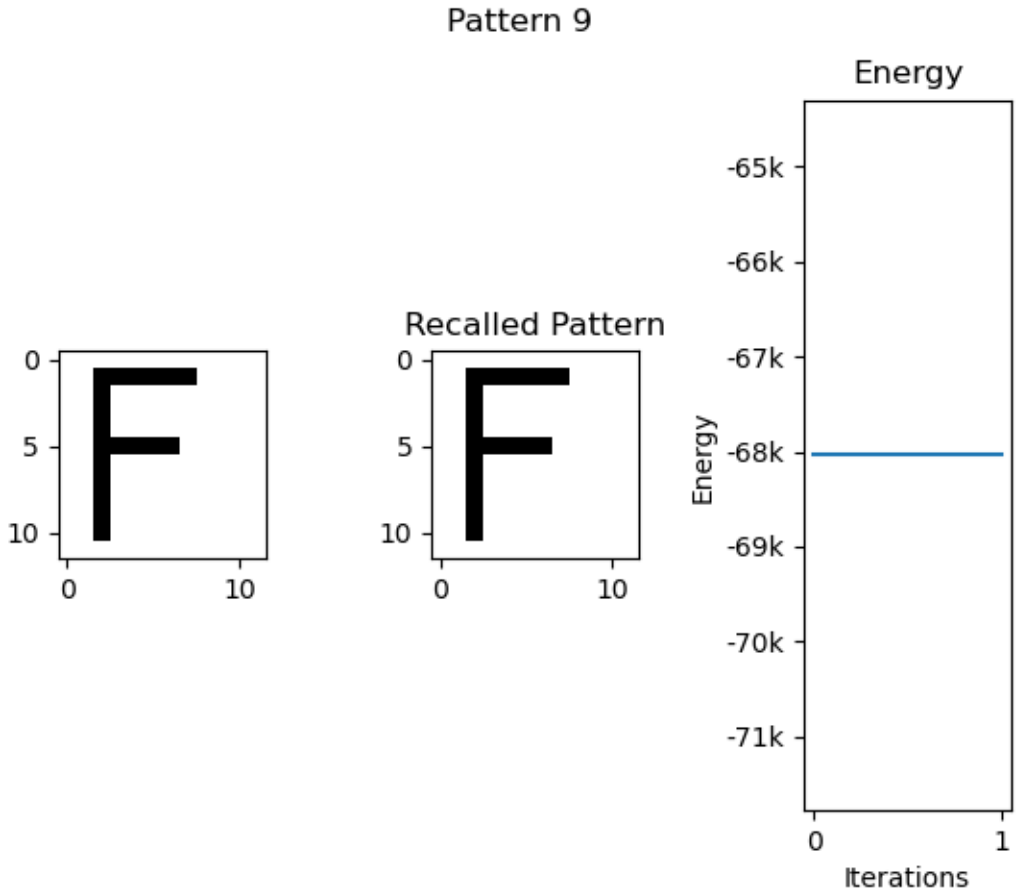

Pattern 7
Recalled Pattern
Energy
0
5
10
0
10
-58k
-60k
-62k
-64k
-66k
-68k
Energy
0
100
Iterations

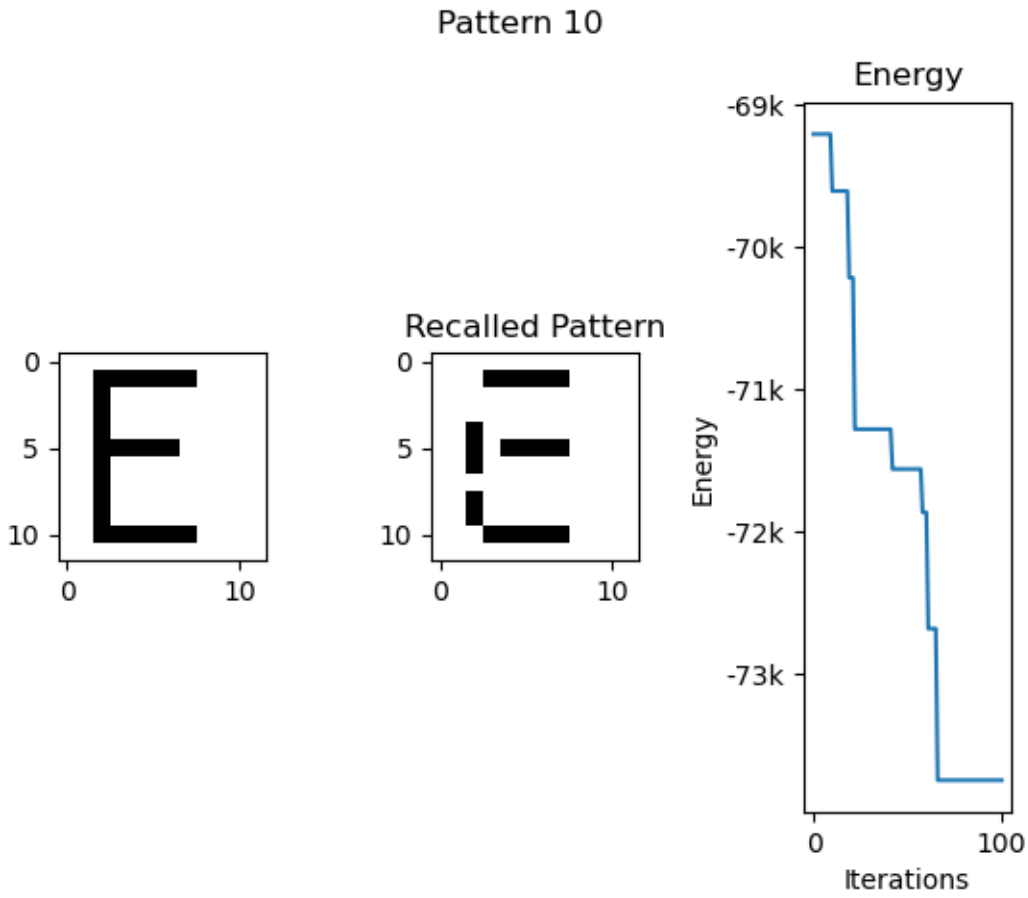

Pattern 7
Recalled Pattern
Energy
0
5
10
0
10
-55k
-56k
-57k
-58k
-59k
-60k
Energy
0
1
Iterations

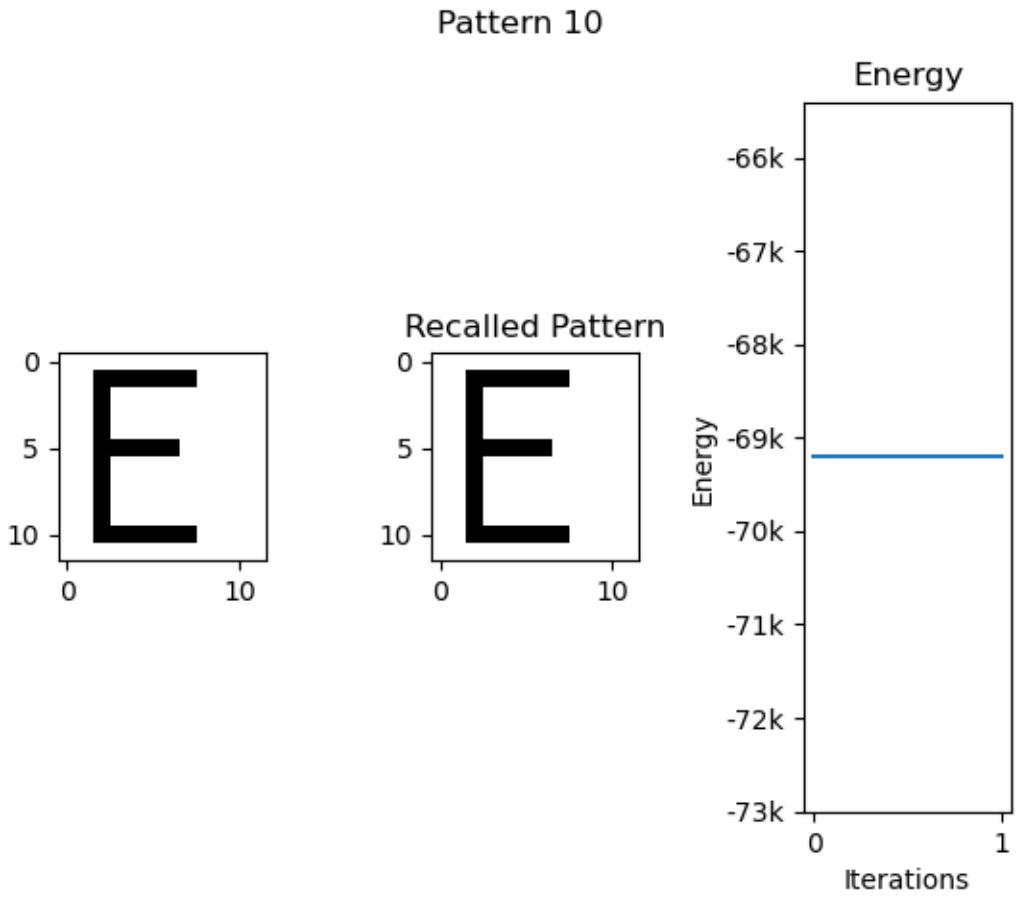

Pattern 8
Recalled Pattern
Energy
0
5
10
0
10
-40k
-42k
-44k
-46k
-48k
-50k
-52k
-54k
-56k
Energy
0
100
Iterations

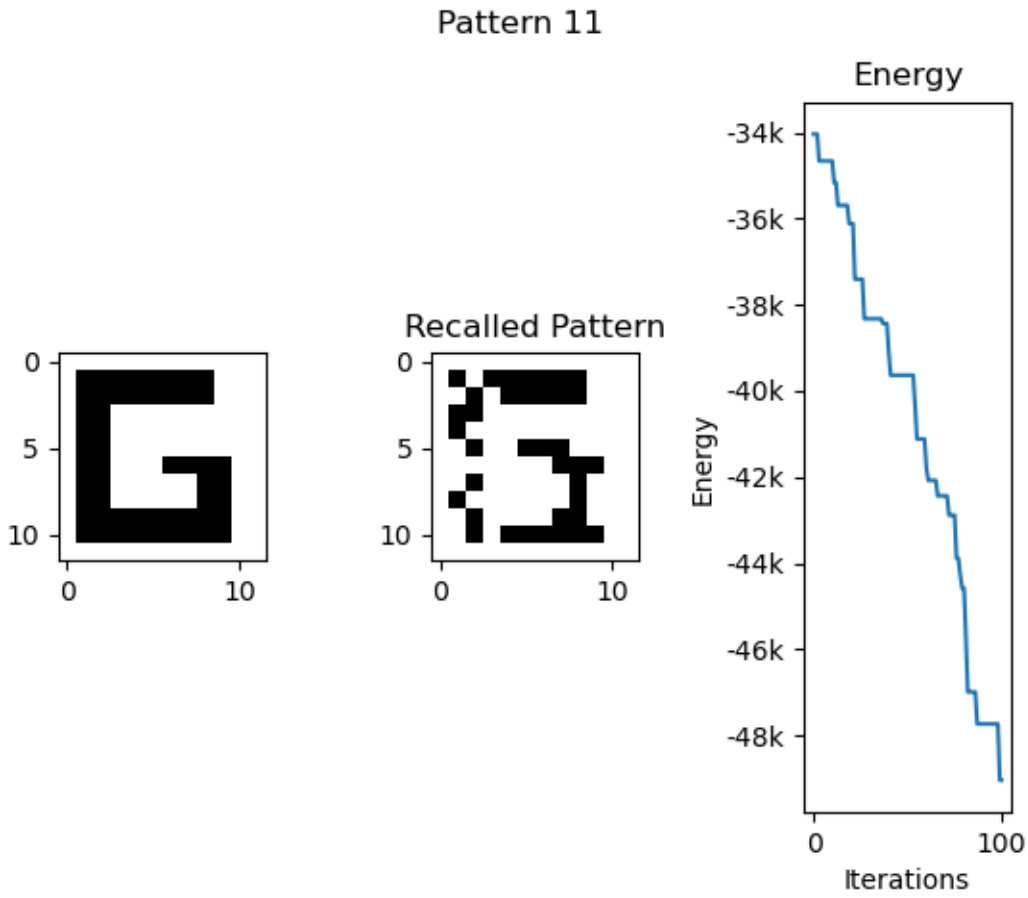

Pattern 8
Recalled Pattern
Energy
0
5
10
0
10
-37k
-37k
-38k
-38k
-39k
-39k
-40k
-40k
-41k
Energy
0
1
Iterations

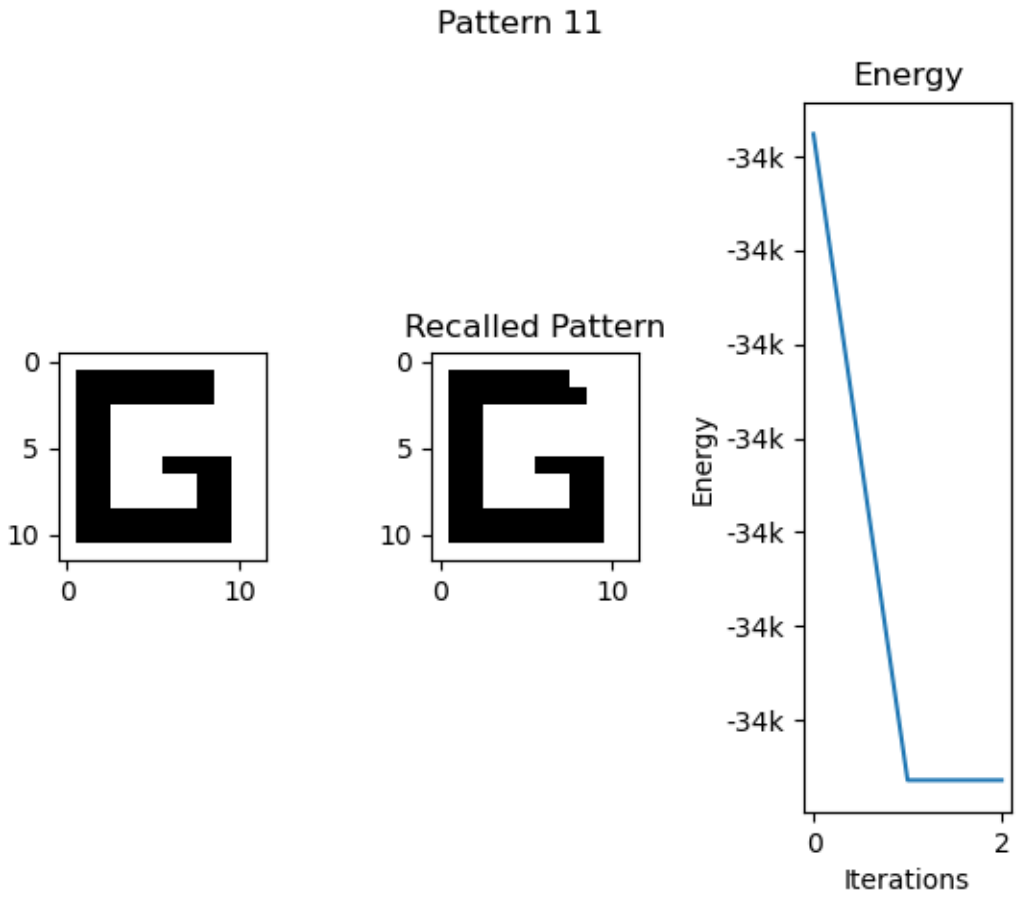

Pattern 9
Recalled Pattern
Energy
0
5
10
0
10
-68k
-70k
-72k
-74k
-76k
Energy
0
100
Iterations

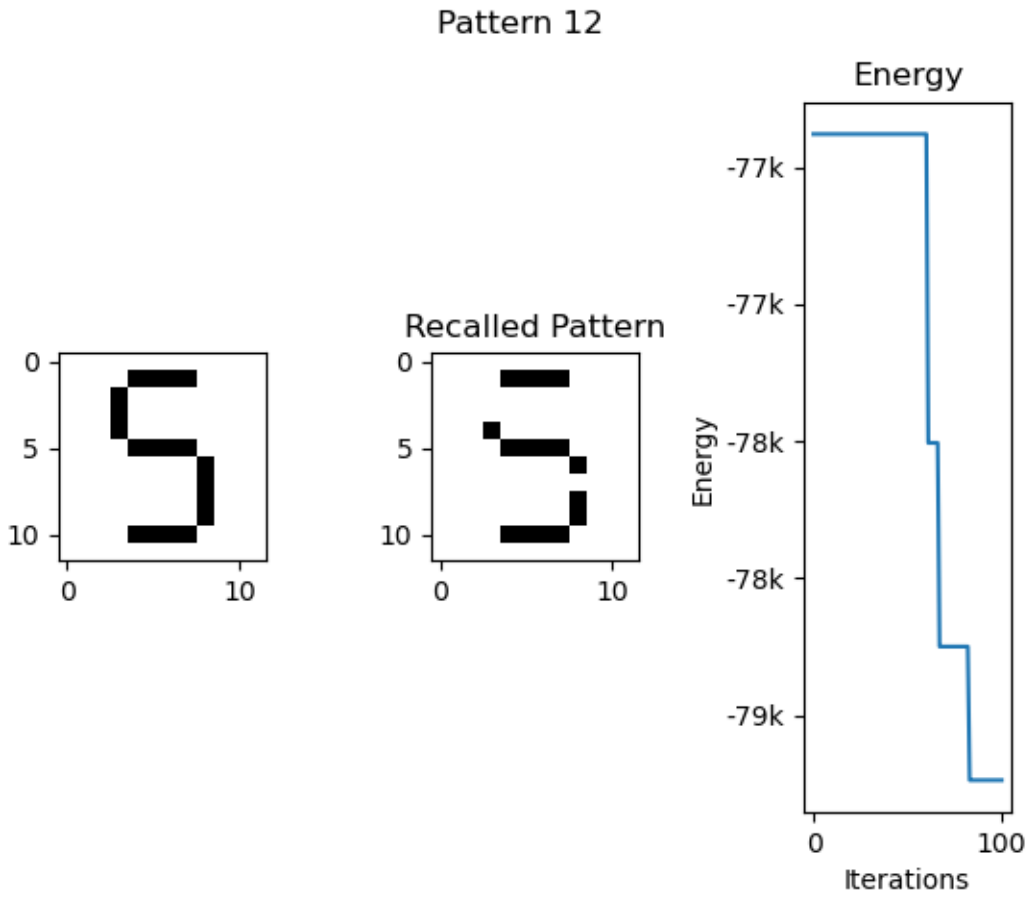

Pattern 9
Recalled Pattern
Energy
0
5
10
0
10
-65k
-66k
-67k
-68k
-69k
-70k
-71k
Energy
0
1
Iterations

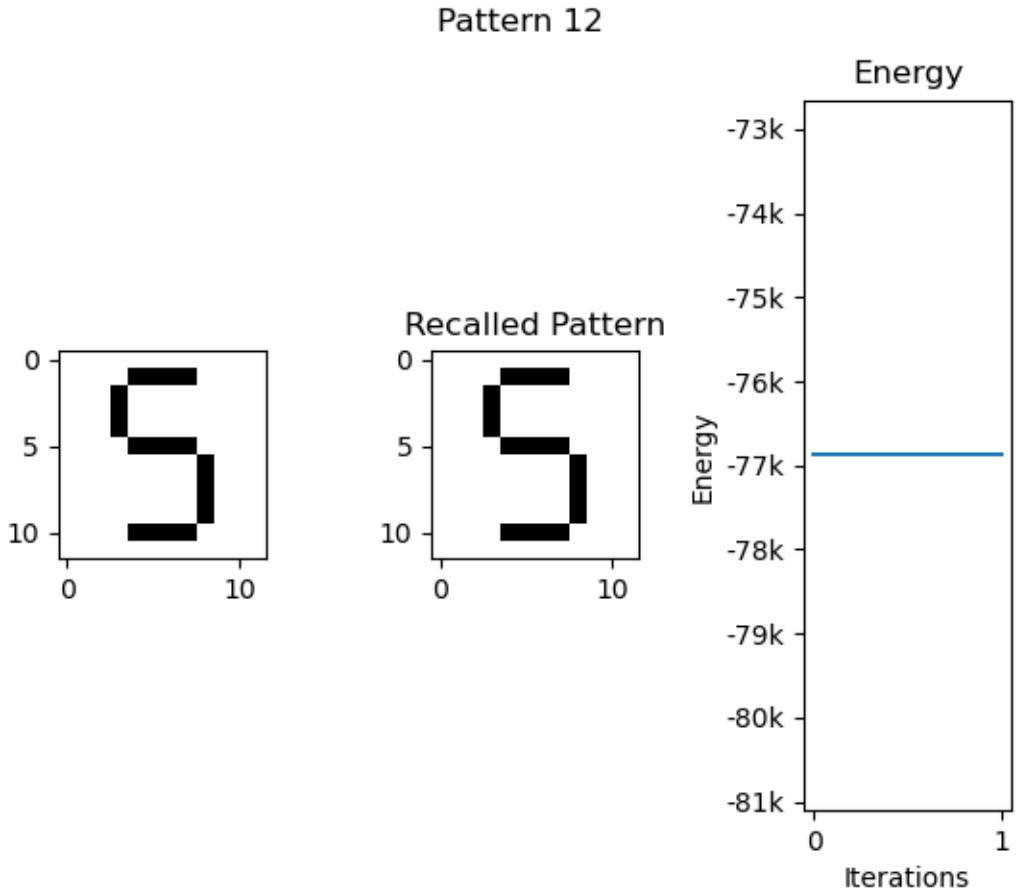

Pattern 10
Recalled Pattern
Energy
0
5
10
0
10
-69k
-70k
-71k
-72k
-73k
Energy
0
100
Iterations

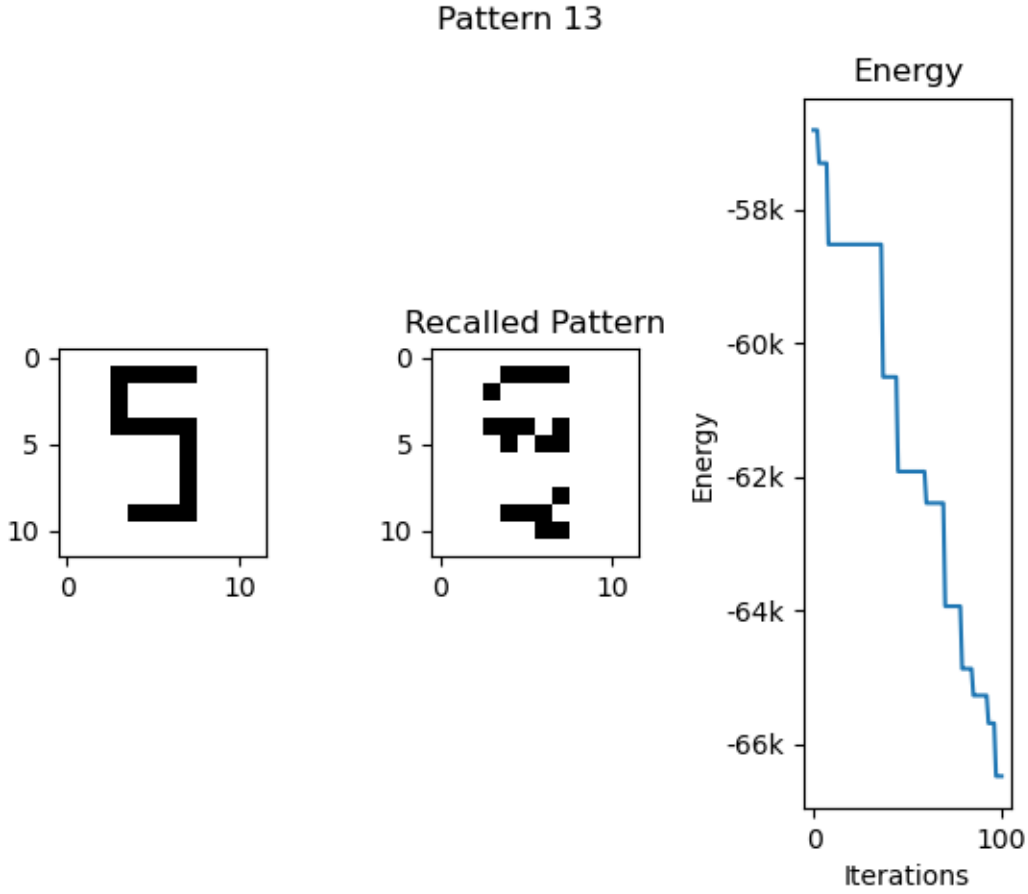

Pattern 10
Recalled Pattern
Energy
0
5
10
0
10
-66k
-67k
-68k
-69k
-70k
-71k
-72k
-73k
Energy
0
1
Iterations

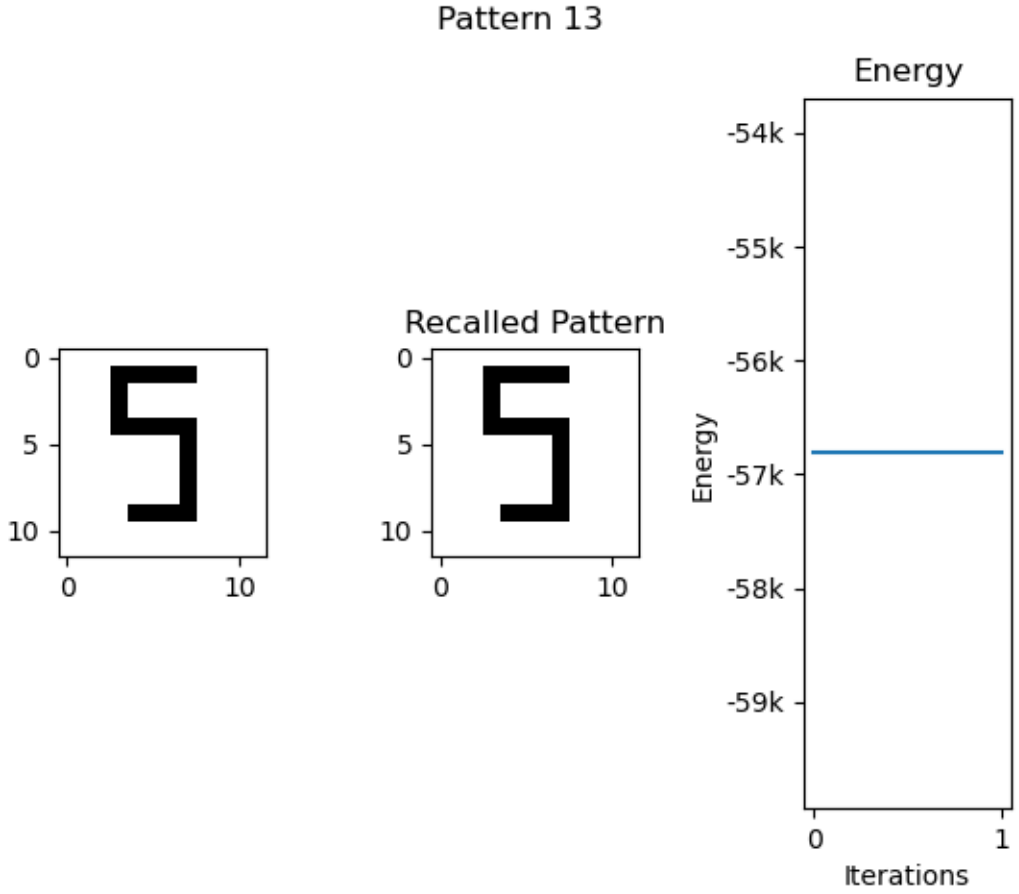

Pattern 11
Recalled Pattern
Energy
0
5
10
0
10
-34k
-36k
-38k
-40k
-42k
-44k
-46k
-48k
Energy
0
100
Iterations

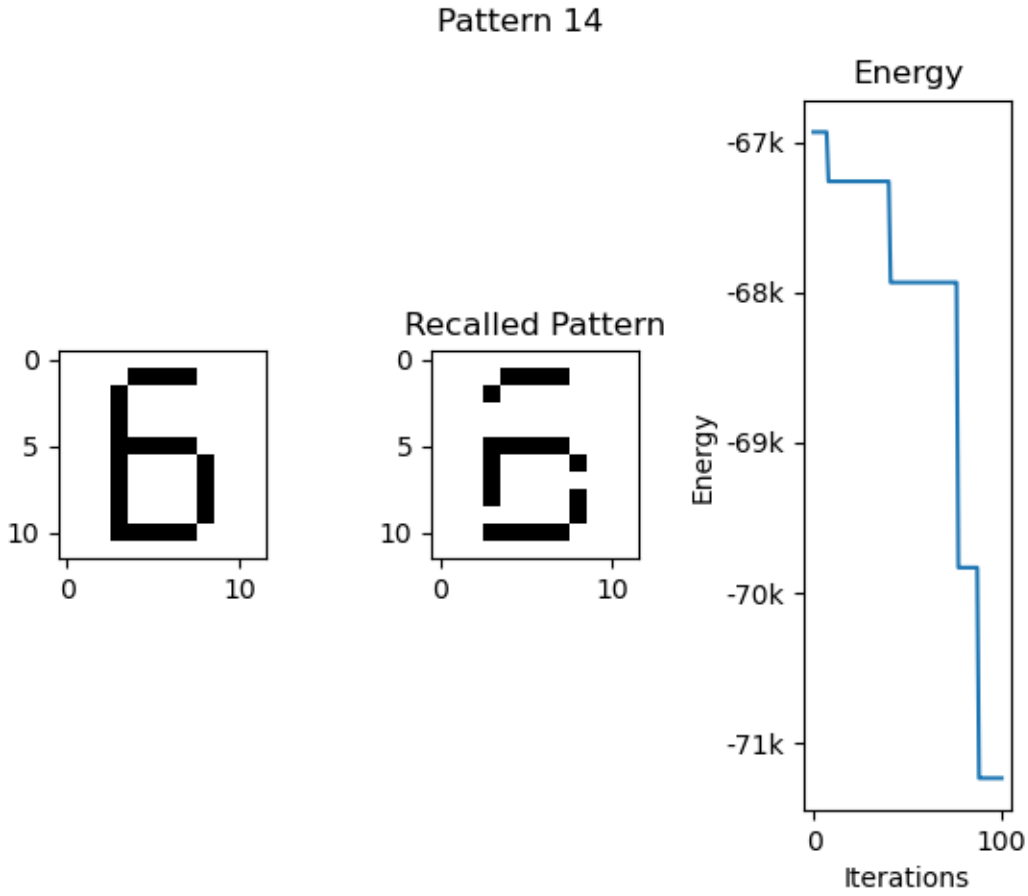

Pattern 11
Recalled Pattern
Energy
0
5
10
0
10
-34k
-34k
-34k
-34k
-34k
-34k
Energy
0
2
Iterations

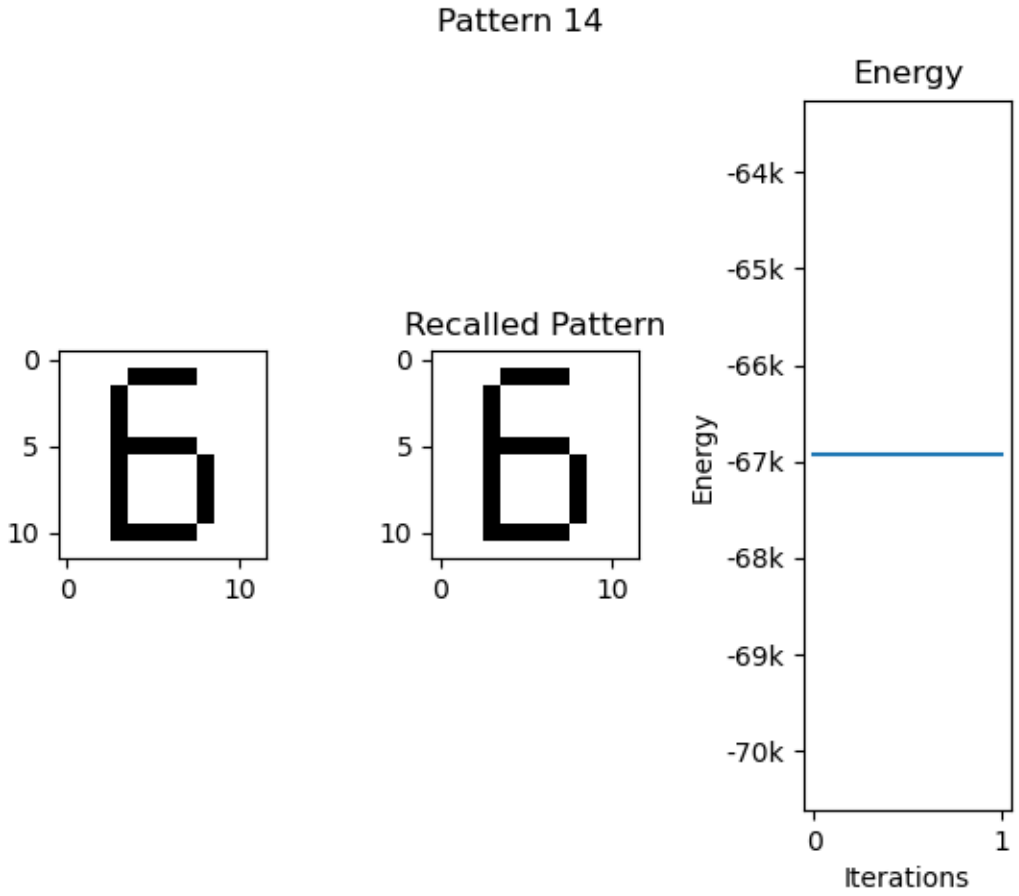

Pattern 12
Recalled Pattern
Energy
0
5
10
0
10
-77k
-77k
-78k
-78k
-79k
Energy
0
100
Iterations

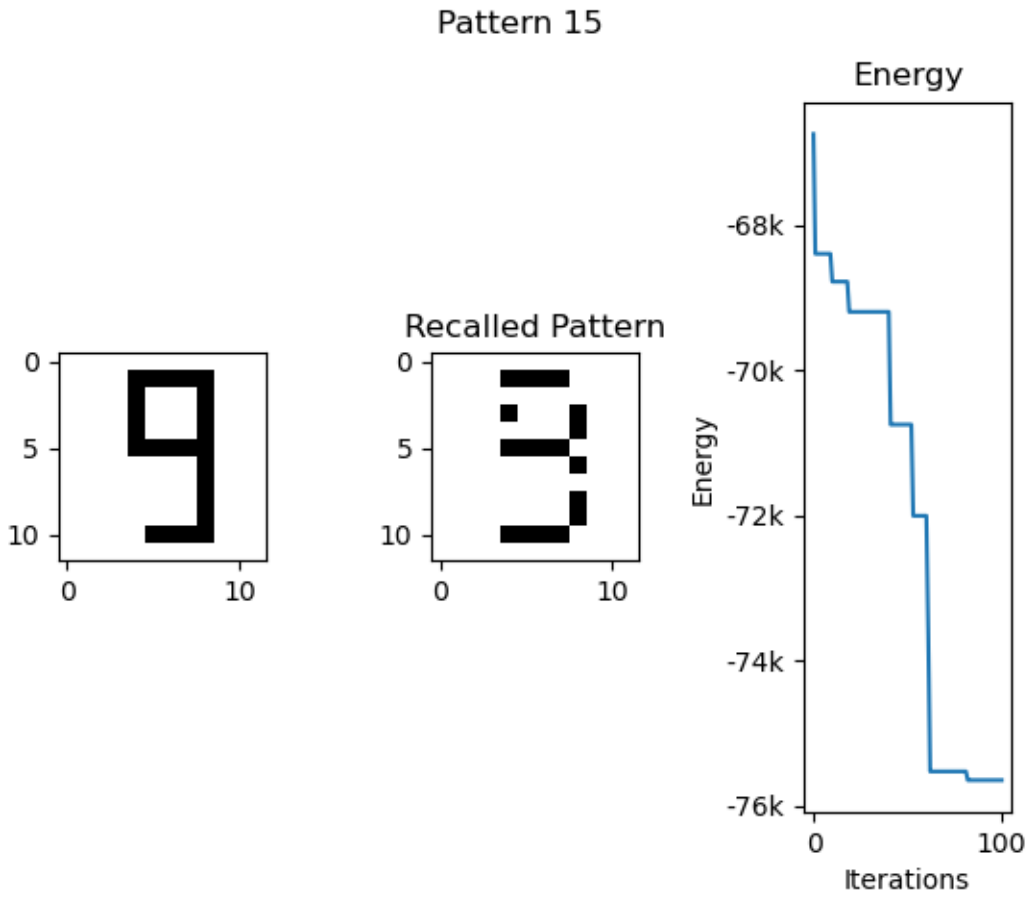

Pattern 12
Recalled Pattern
Energy
0
5
10
0
10
-73k
-74k
-75k
-76k
-77k
-78k
-79k
-80k
-81k
Energy
0
1
Iterations

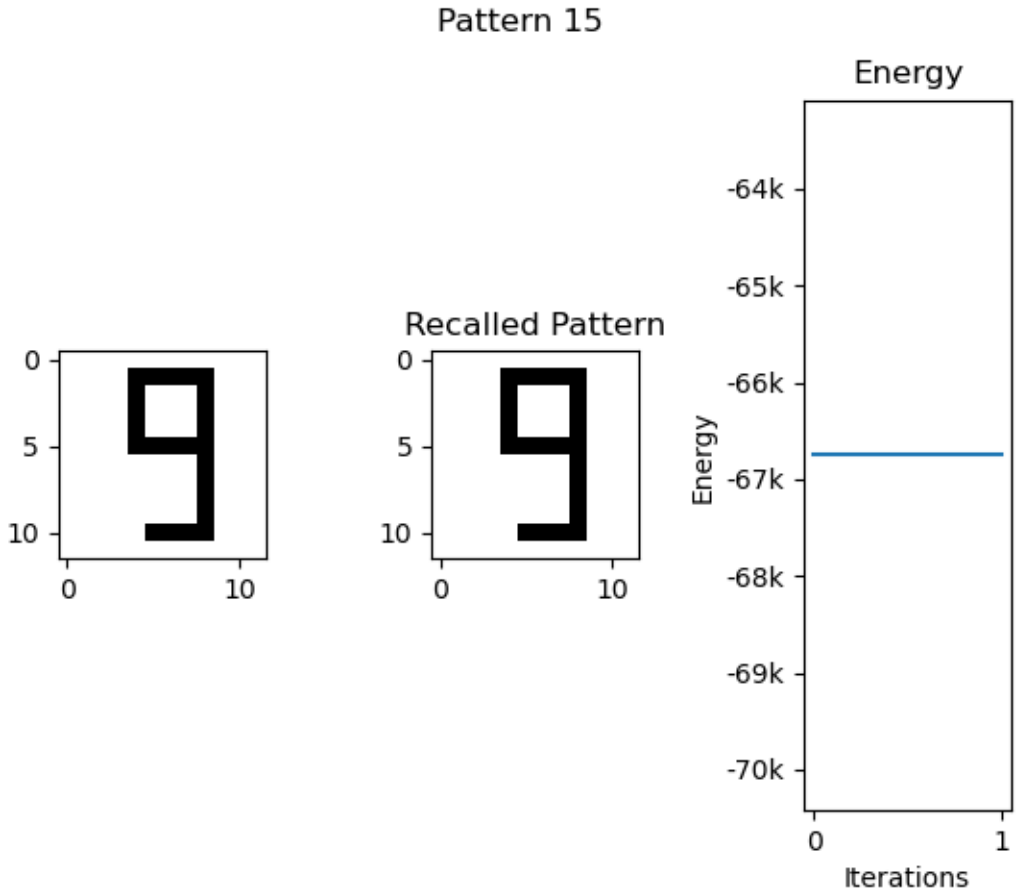

Pattern 13
Recalled Pattern
Energy
0
5
10
0
10
-58k
-60k
-62k
-64k
-66k
Energy
0
100
Iterations

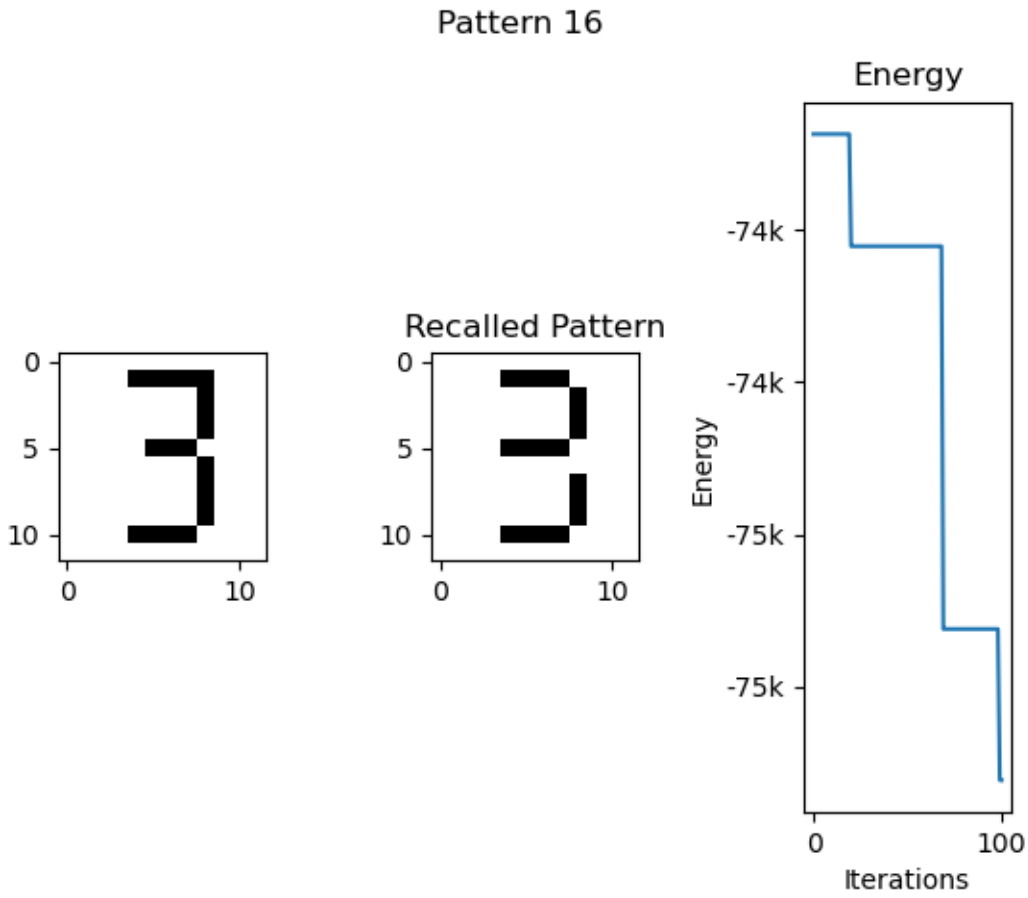

Pattern 13
Recalled Pattern
Energy
0
5
10
0
10
-54k
-55k
-56k
-57k
-58k
-59k
Energy
0
1
Iterations

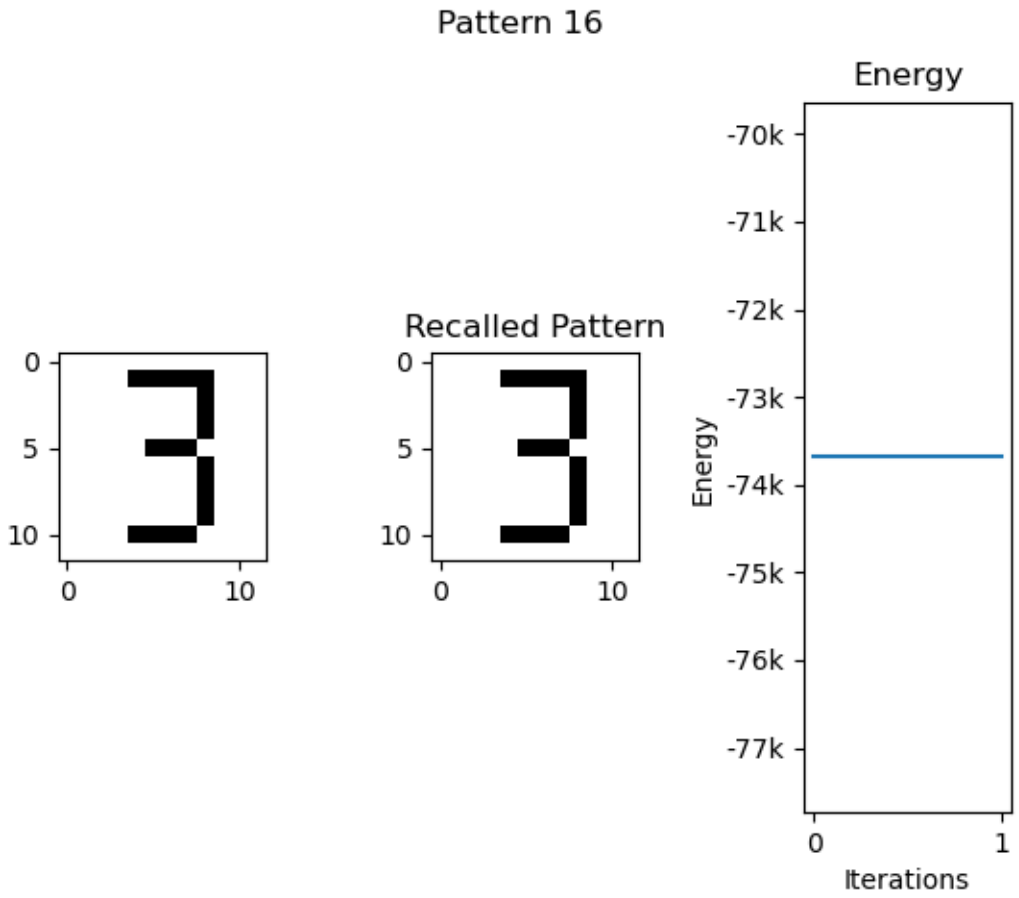

Pattern 14
Recalled Pattern
Energy
0
5
10
0
10
-67k
-68k
-69k
-70k
-71k
Energy
0
100
Iterations

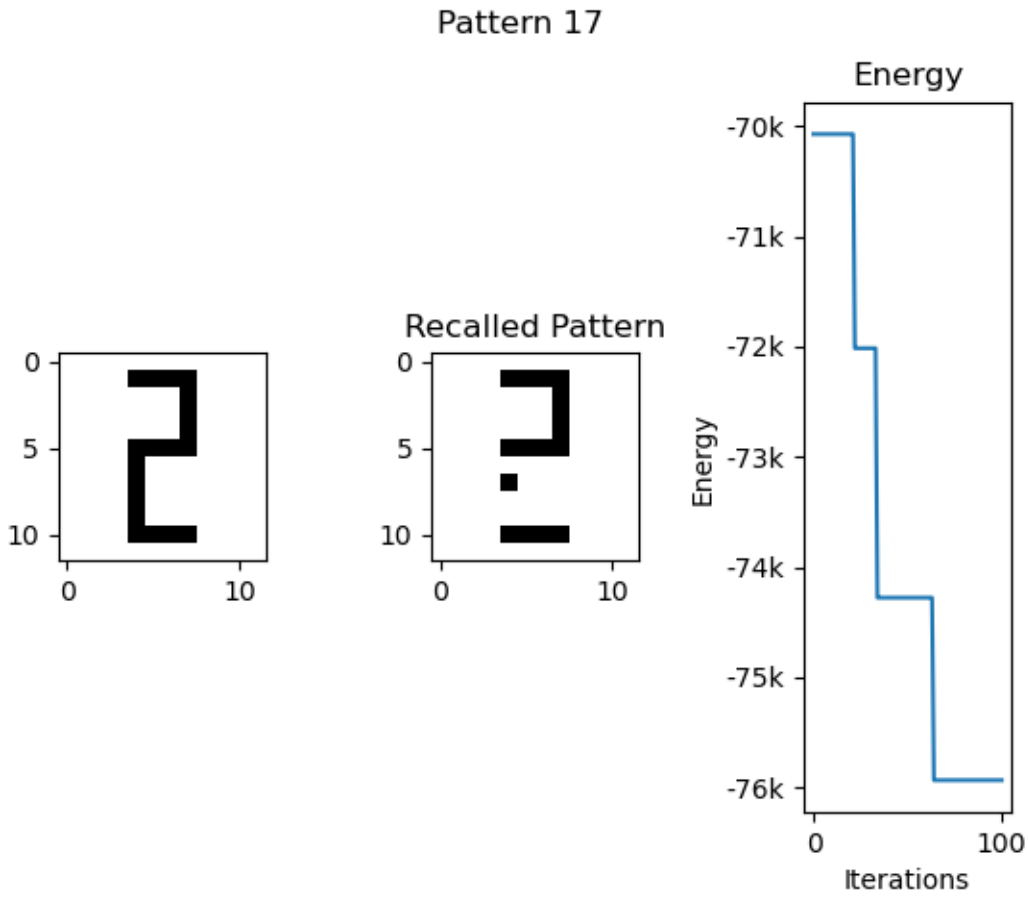

Pattern 14
Recalled Pattern
Energy
0
5
10
0
10
-64k
-65k
-66k
-67k
-68k
-69k
-70k
Energy
0
1
Iterations

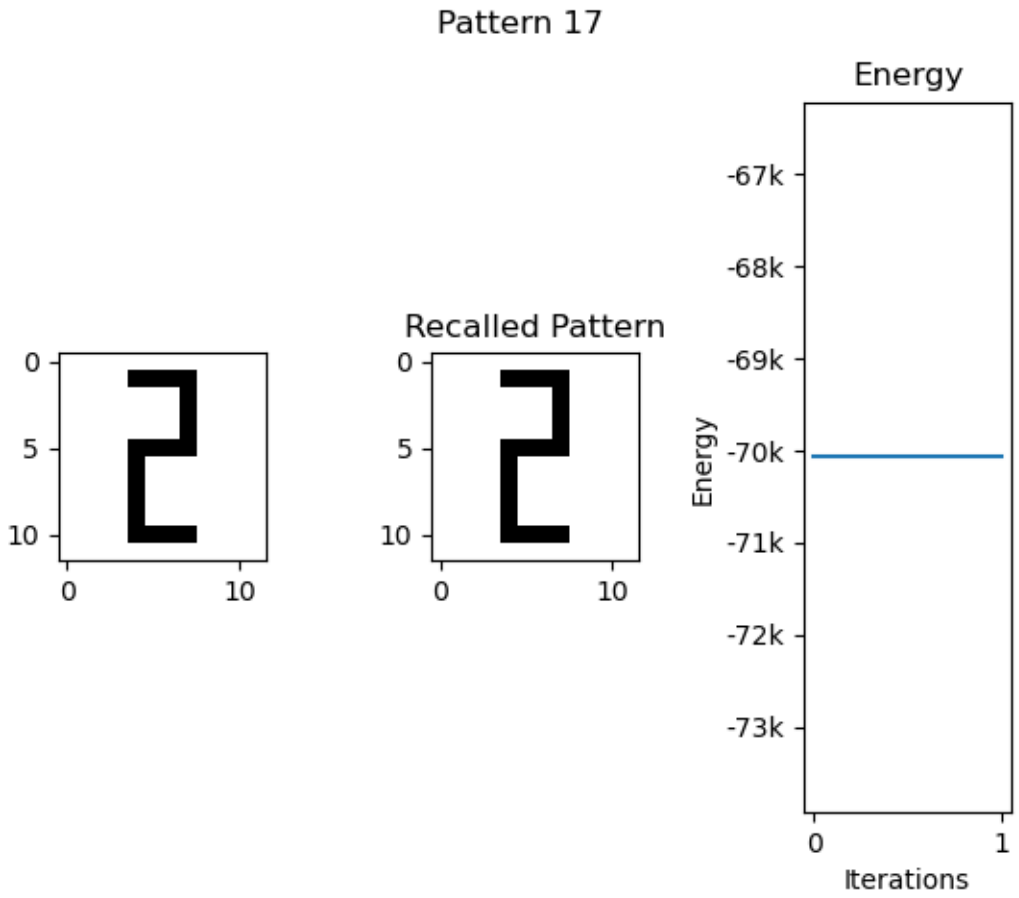

Pattern 15
Recalled Pattern
Energy
0
5
10
0
10
-68k
-70k
-72k
-74k
-76k
Energy
0
100
Iterations

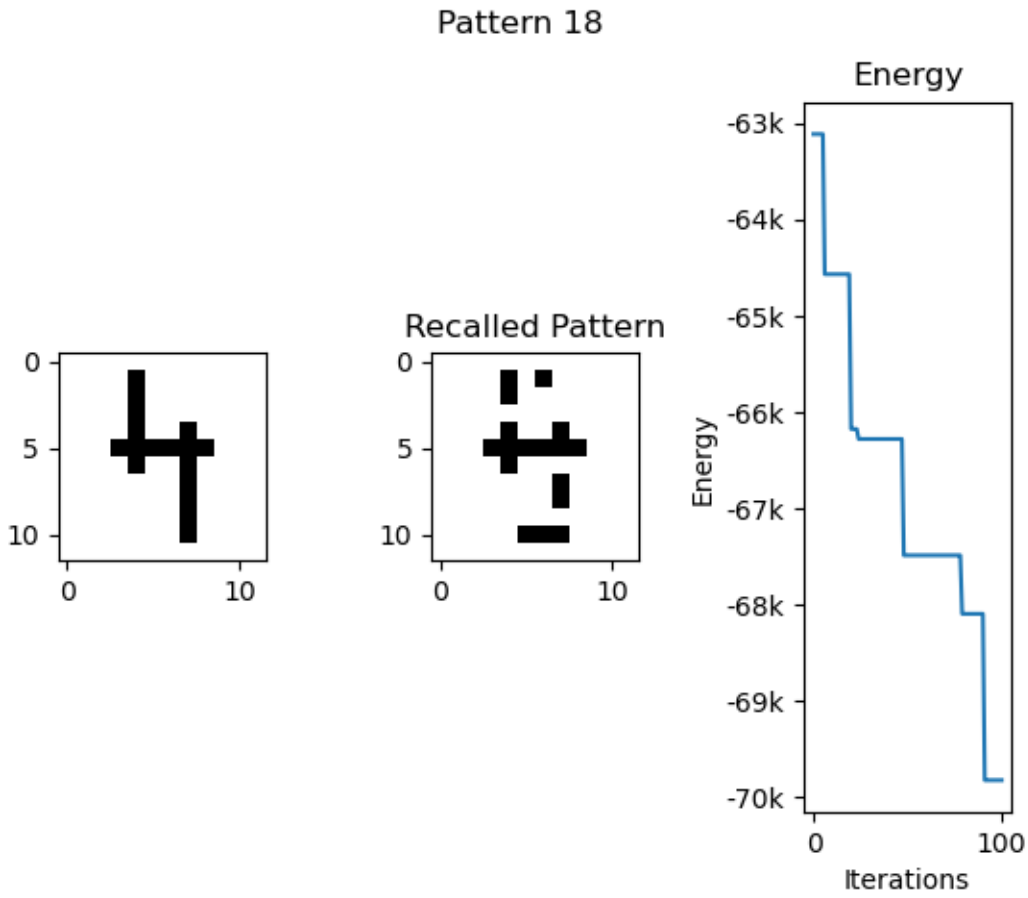

Pattern 15
Recalled Pattern
Energy
0
5
10
0
10
-64k
-65k
-66k
-67k
-68k
-69k
-70k
Energy
0
1
Iterations

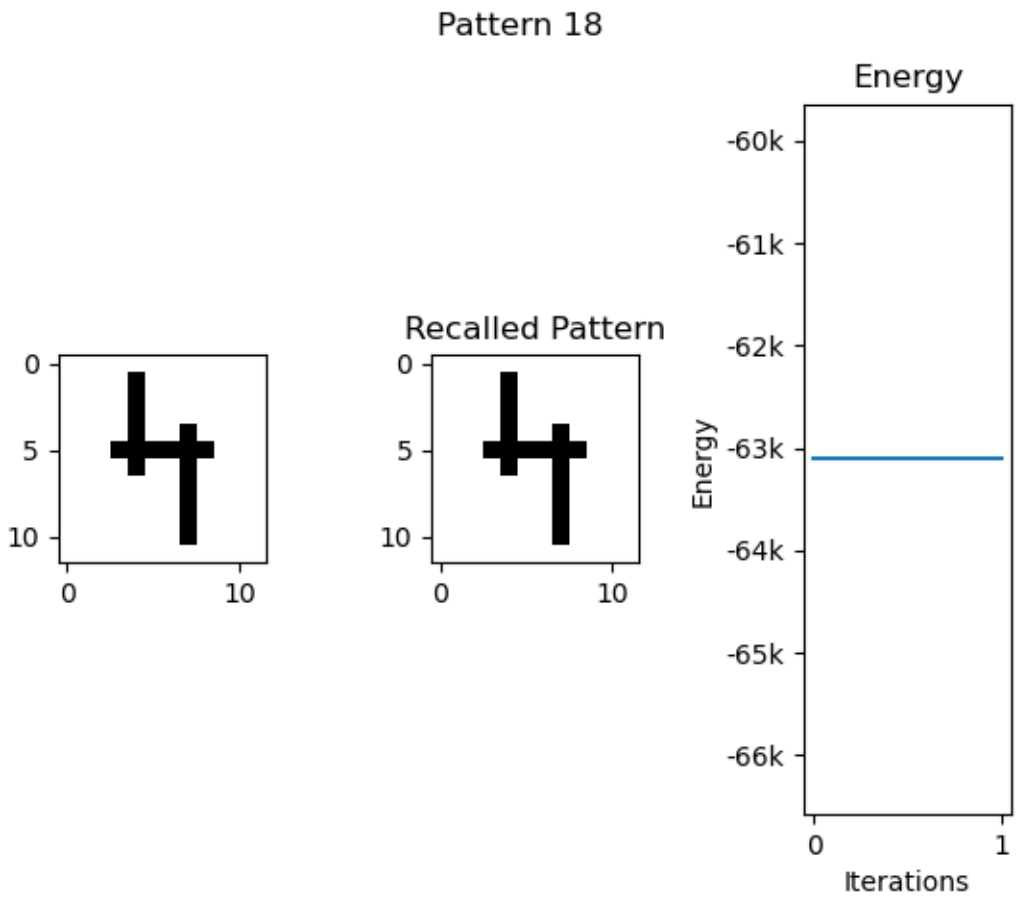

Pattern 16
Recalled Pattern
Energy
0
5
10
0
10
-74k
-74k
-75k
-75k
Energy
0
100
Iterations


Pattern 16
Recalled Pattern
Energy
0
5
10
0
10
-70k
-71k
-72k
-73k
-74k
-75k
-76k
-77k
Energy
0
1
Iterations

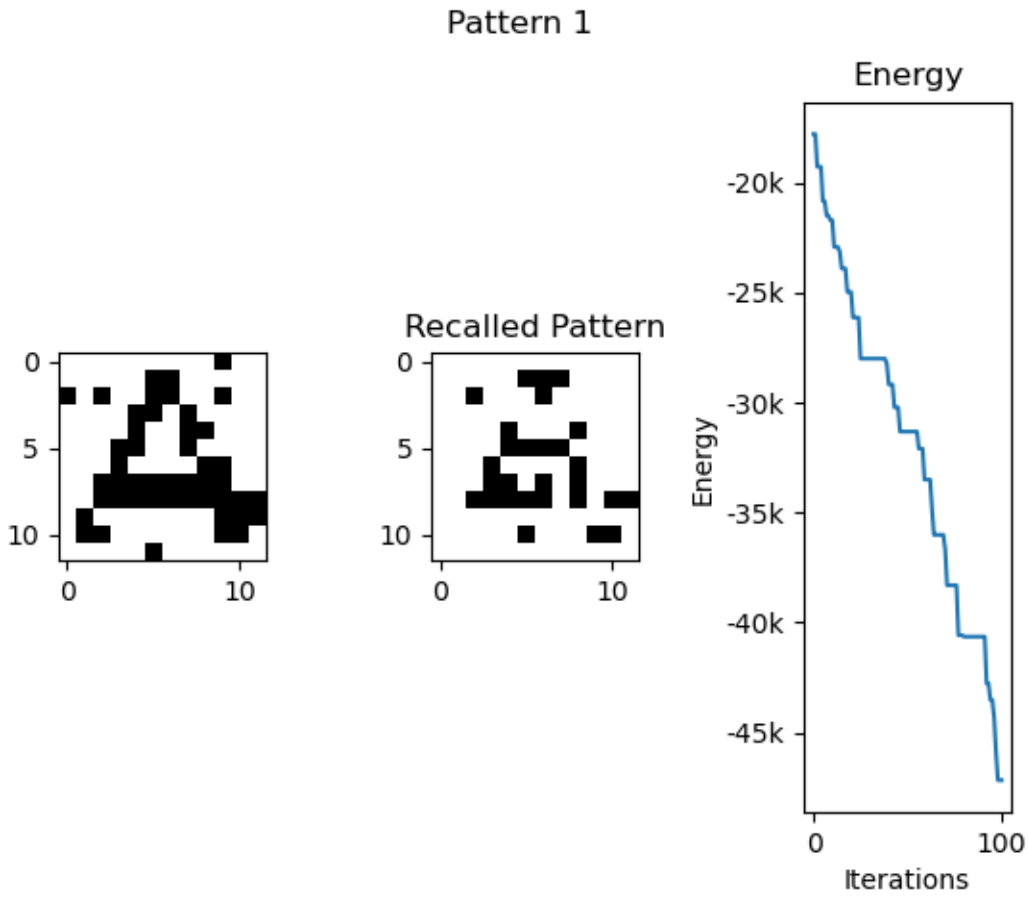

Pattern 17
Recalled Pattern
Energy
0
5
10
0
10
-70k
-71k
-72k
-73k
-74k
-75k
-76k
Energy
0
100
Iterations

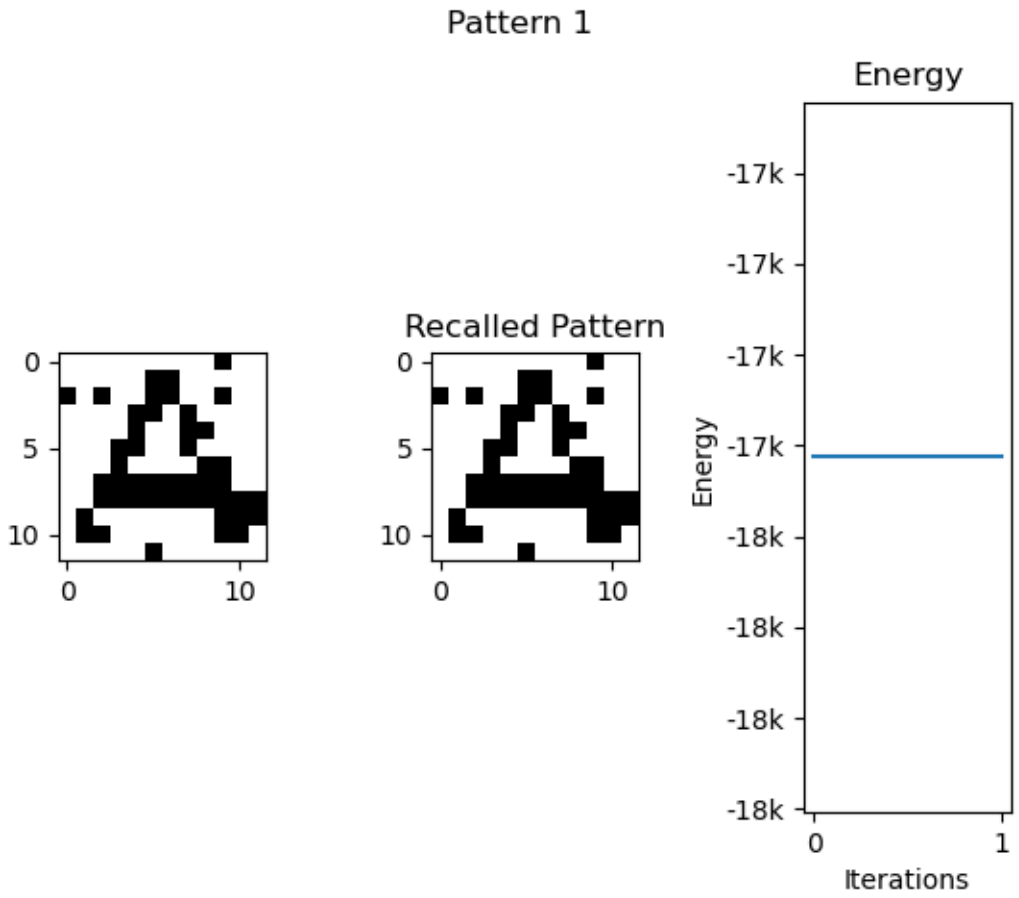

Pattern 17
Recalled Pattern
Energy
0
5
10
0
10
-67k
-68k
-69k
-70k
-71k
-72k
-73k
Energy
0
1
Iterations

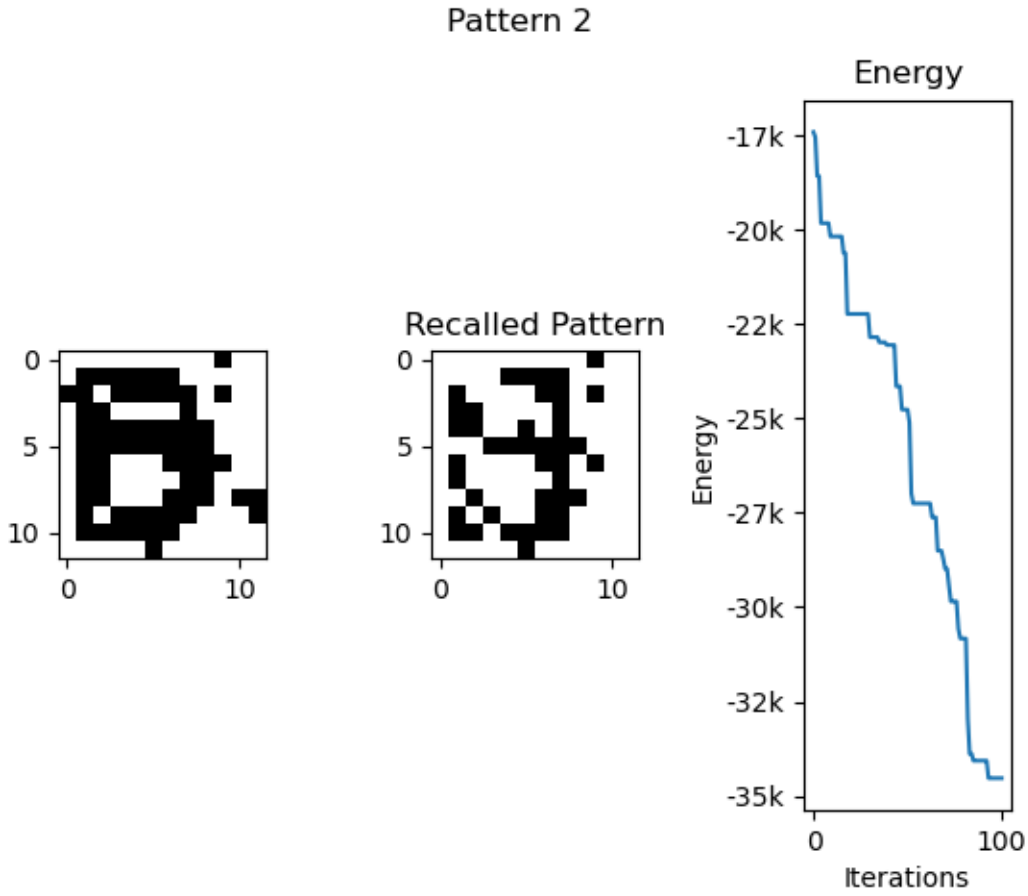

Pattern 18
Recalled Pattern
Energy
0
5
10
0
10
-63k
-64k
-65k
-66k
-67k
-68k
-69k
-70k
Energy
0
100
Iterations

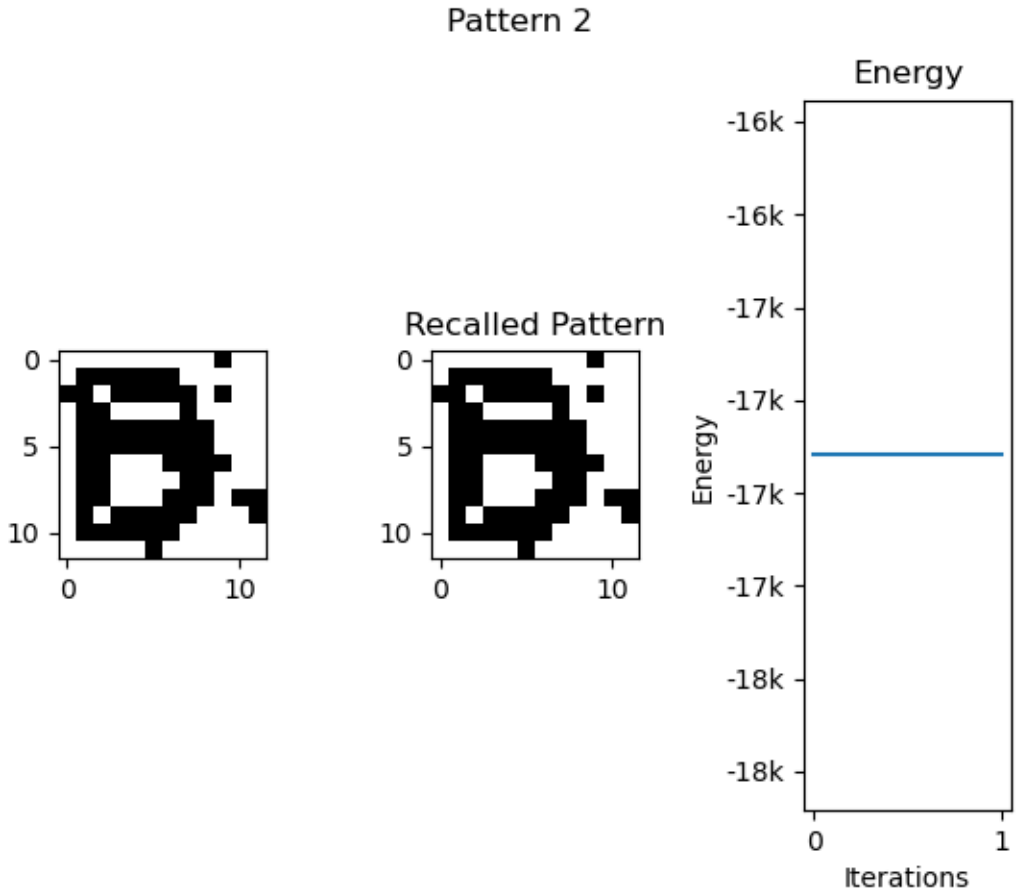


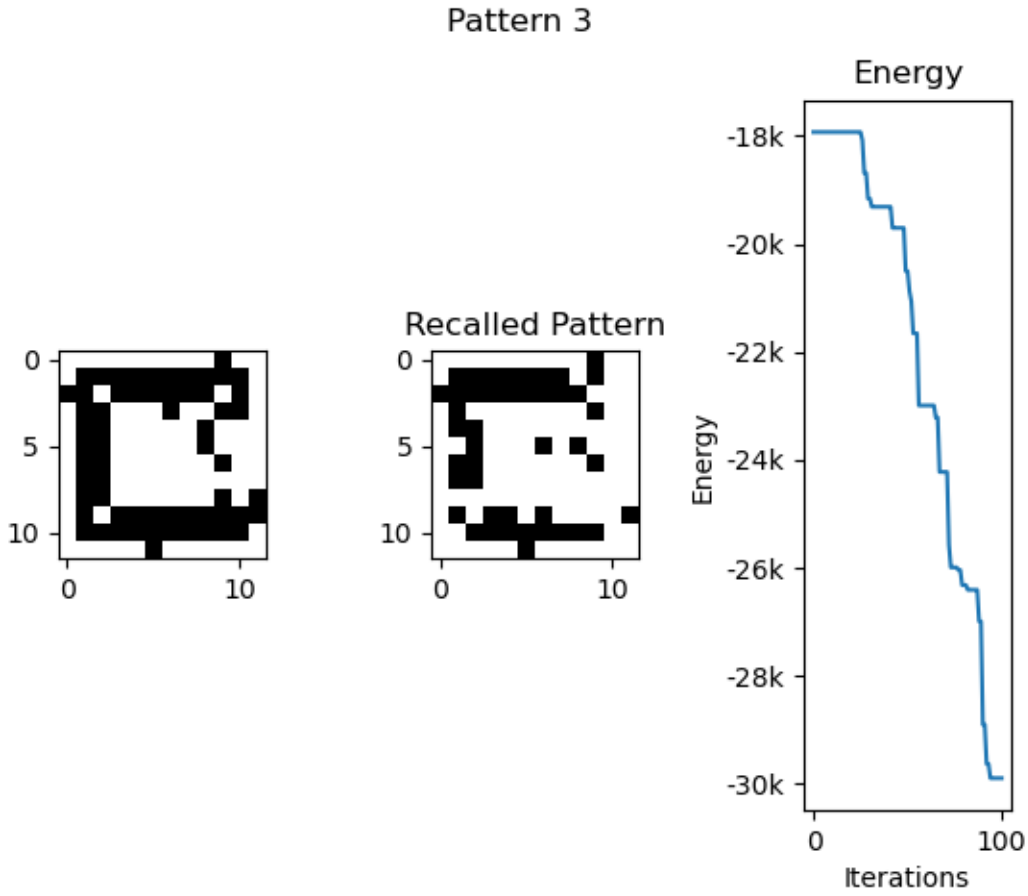
Next comes images of the test data.

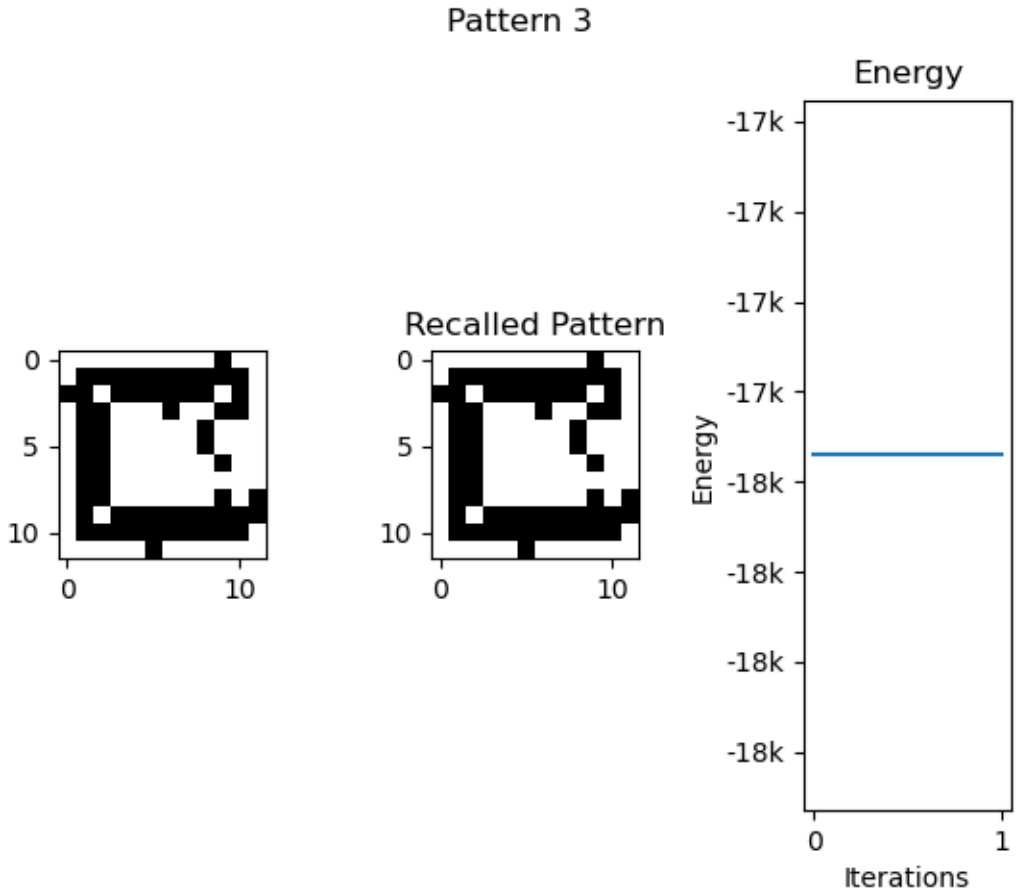

Pattern 1
Recalled Pattern
Energy
0
5
10
0
10
-20k
-25k
-30k
-35k
-40k
-45k
Energy
0
100
Iterations

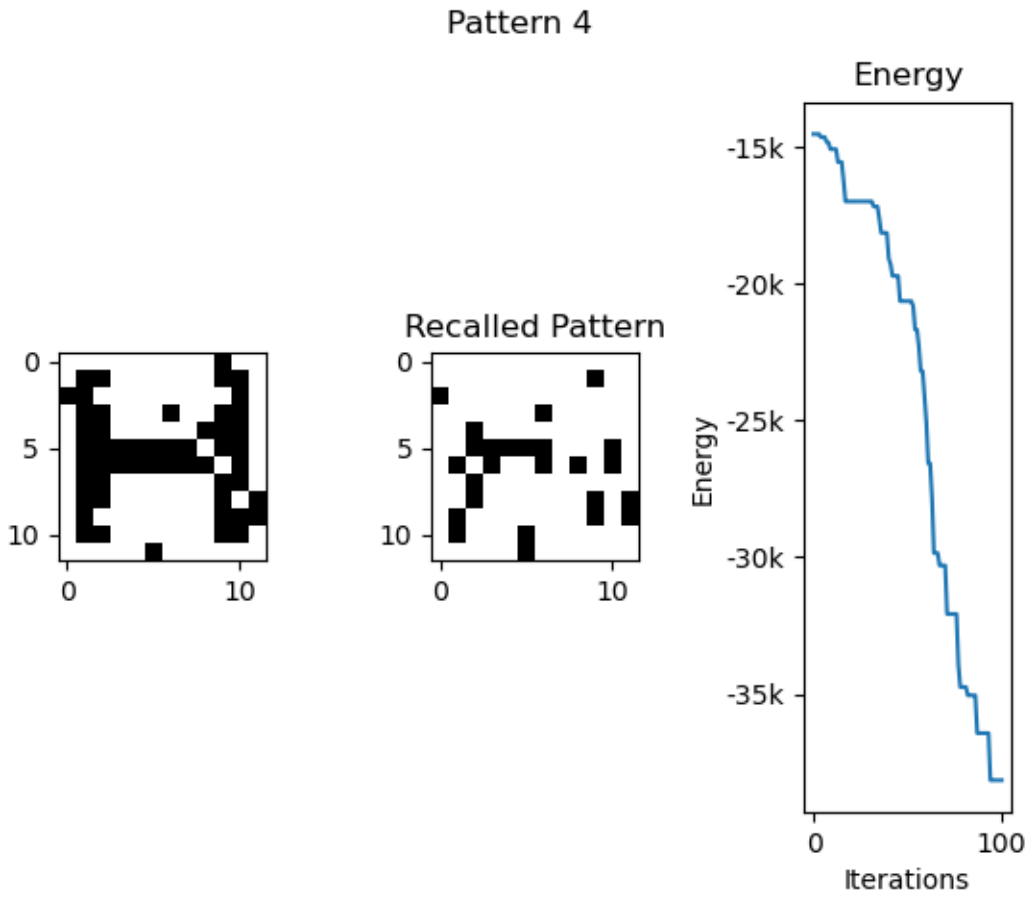

Pattern 1
Recalled Pattern
Energy
0
5
10
0
10
-17k
-17k
-17k
-17k
-18k
-18k
-18k
-18k
Energy
0
1
Iterations

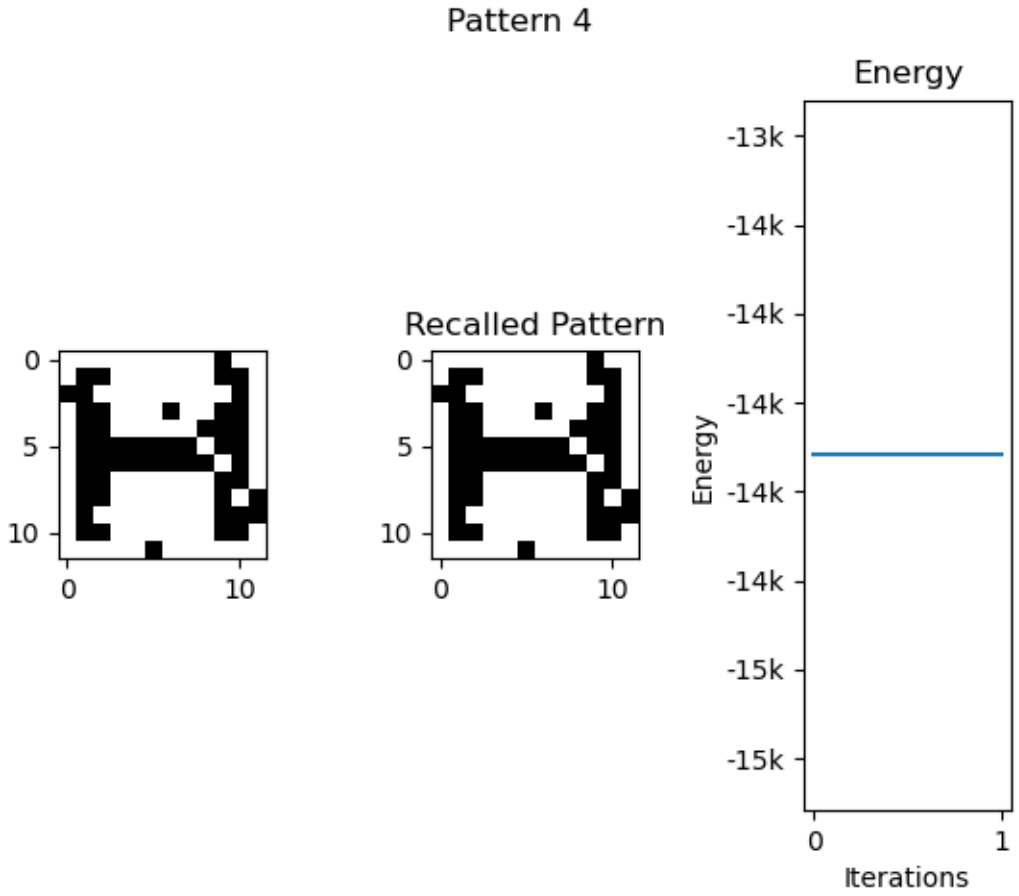

Pattern 2
Recalled Pattern
Energy
0
5
10
0
10
-17k
-20k
-22k
-25k
-27k
-30k
-32k
-35k
0
100
Iterations

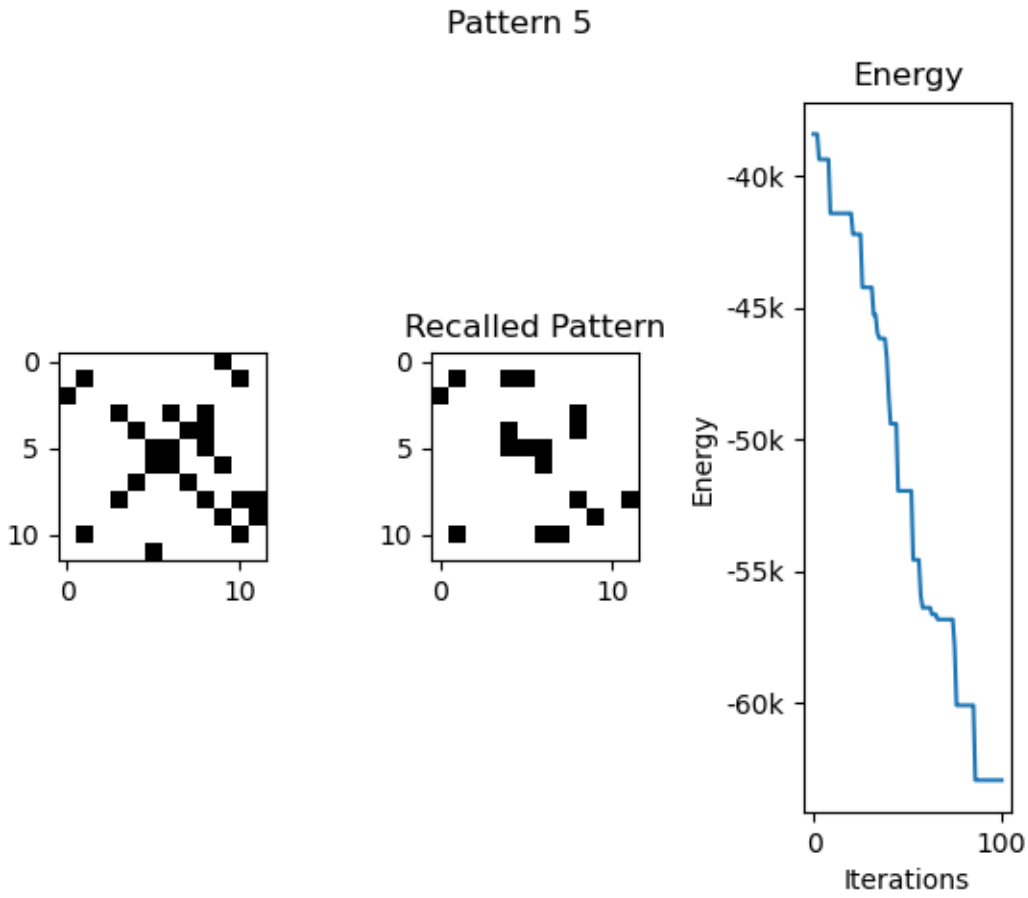

Pattern 2
Recalled Pattern
Energy
0
5
10
0
10
-16k
-16k
-17k
-17k
-17k
-17k
-18k
-18k
Energy
0
1
Iterations

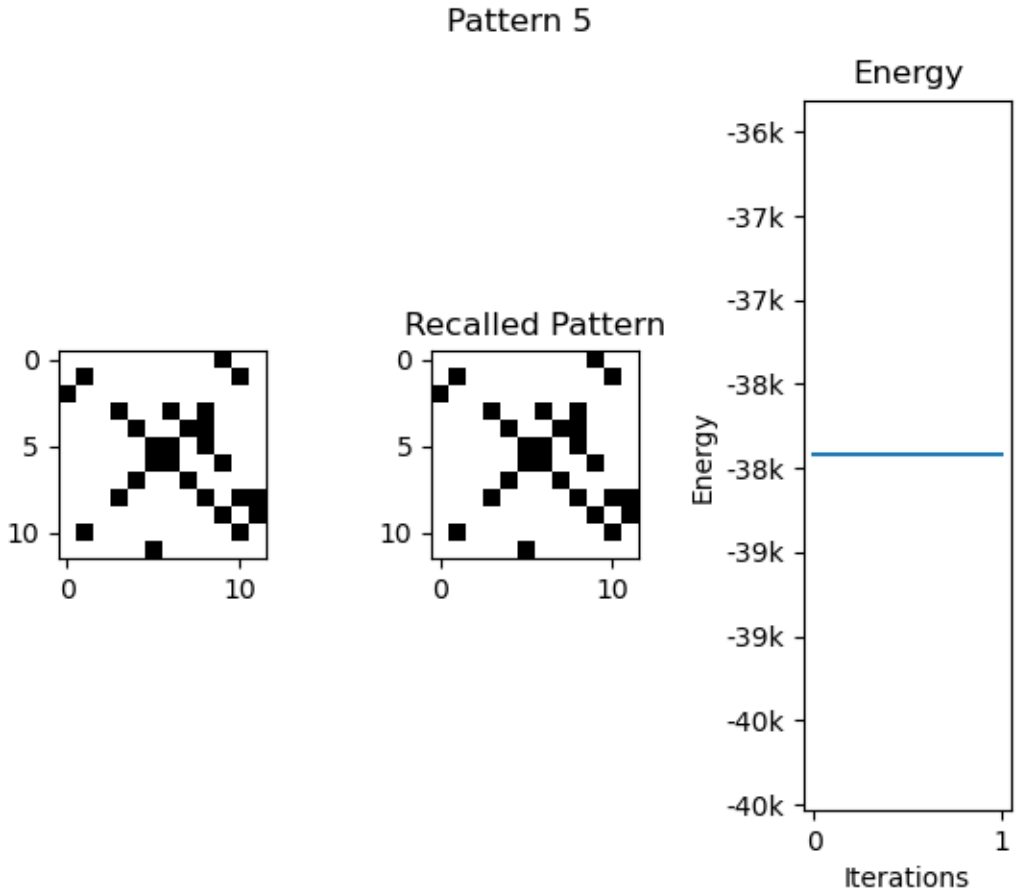

Pattern 3
Recalled Pattern
Energy
0
5
10
0
10
-18k
-20k
-22k
-24k
-26k
-28k
-30k
Energy
0
100
Iterations

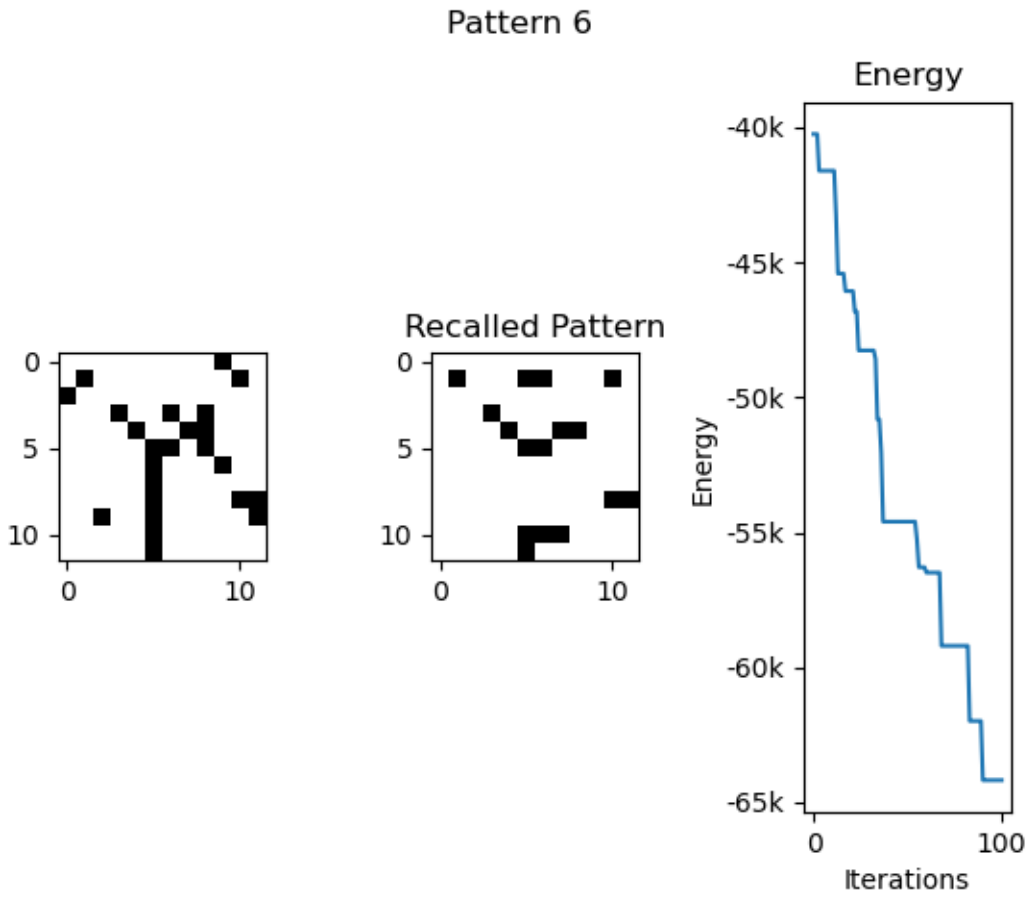

Pattern 3
Recalled Pattern
Energy
0
5
10
0
10
-17k
-17k
-17k
-17k
-18k
-18k
-18k
-18k
Energy
0
1
Iterations

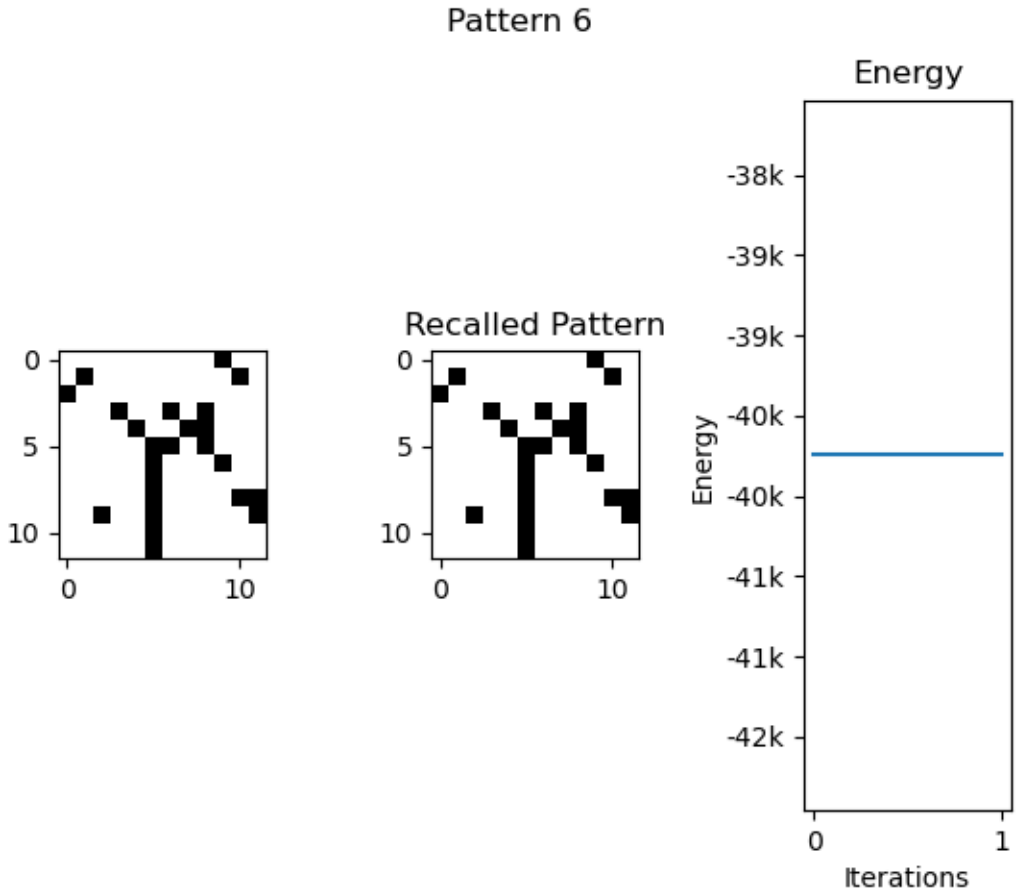

Pattern 4
Recalled Pattern
Energy
0
5
10
0
10
-15k
-20k
-25k
-30k
-35k
Energy
0
100
Iterations

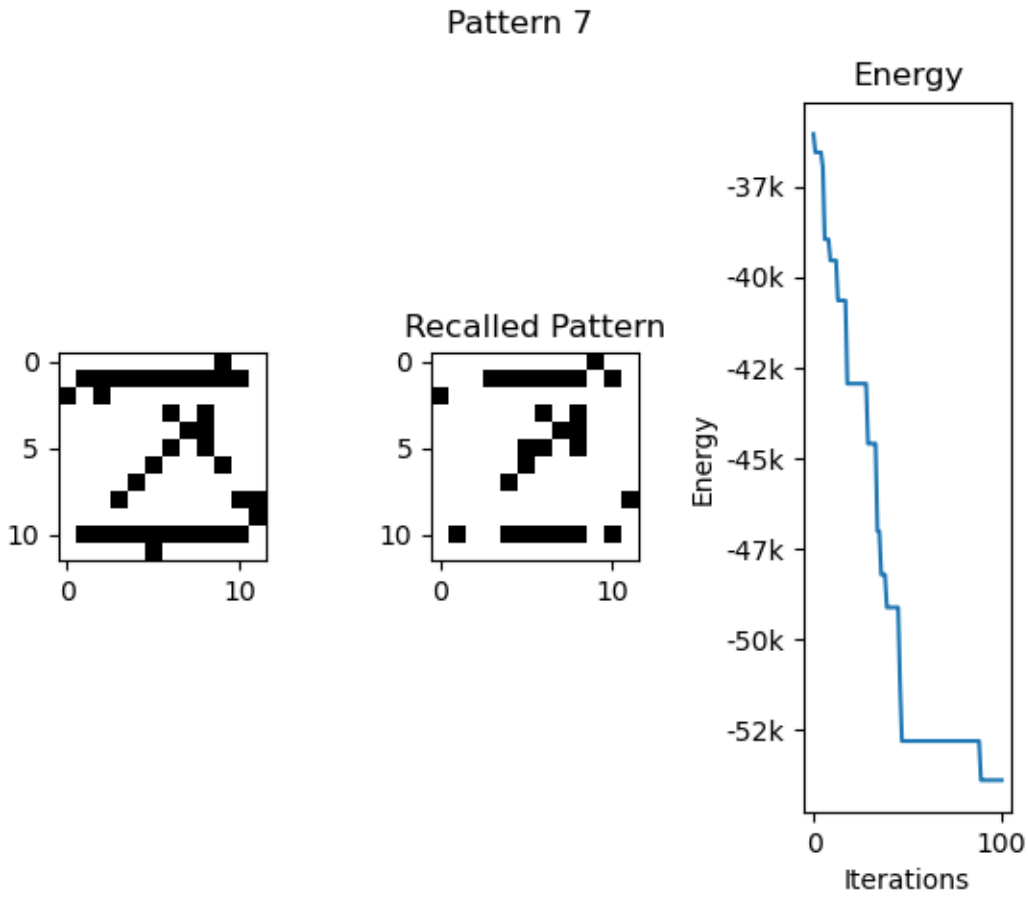

Pattern 4
Recalled Pattern
Energy
0
5
10
0
10
-13k
-14k
-14k
-14k
-14k
-14k
-15k
-15k
Energy
0
1
Iterations

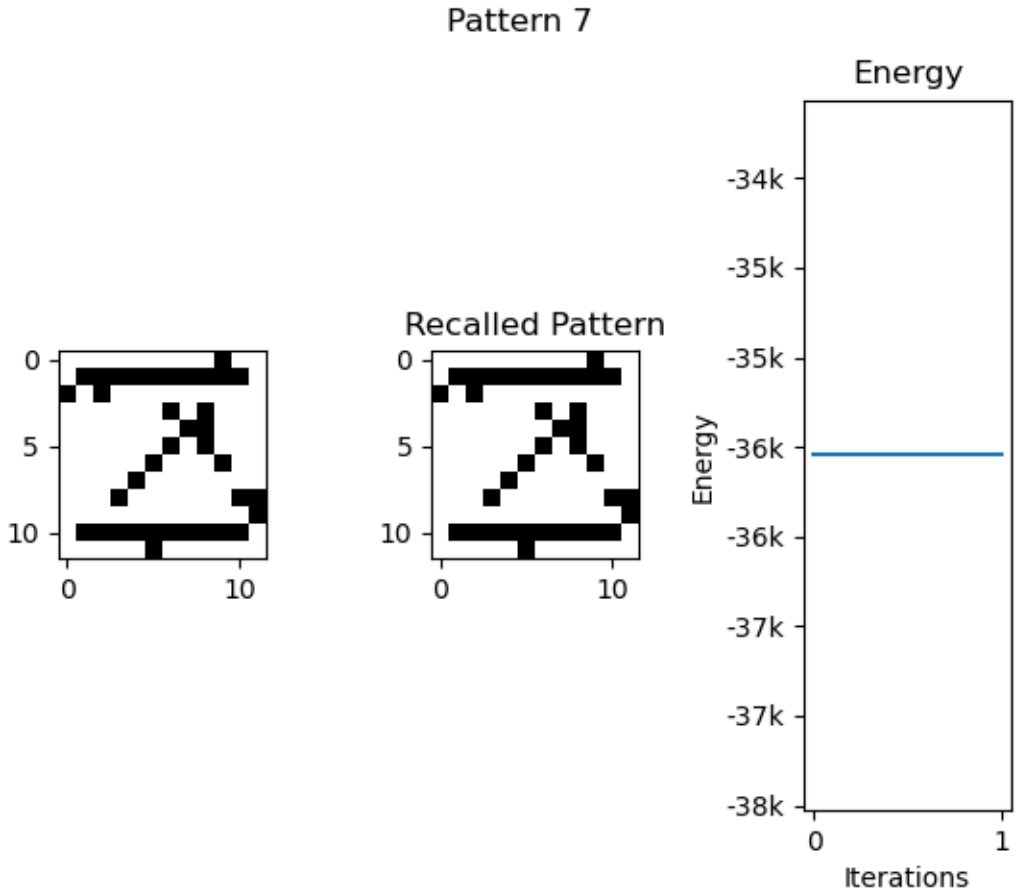

Pattern 5
Recalled Pattern
Energy
0
5
10
0
10
-40k
-45k
-50k
-55k
-60k
Energy
0
100
Iterations

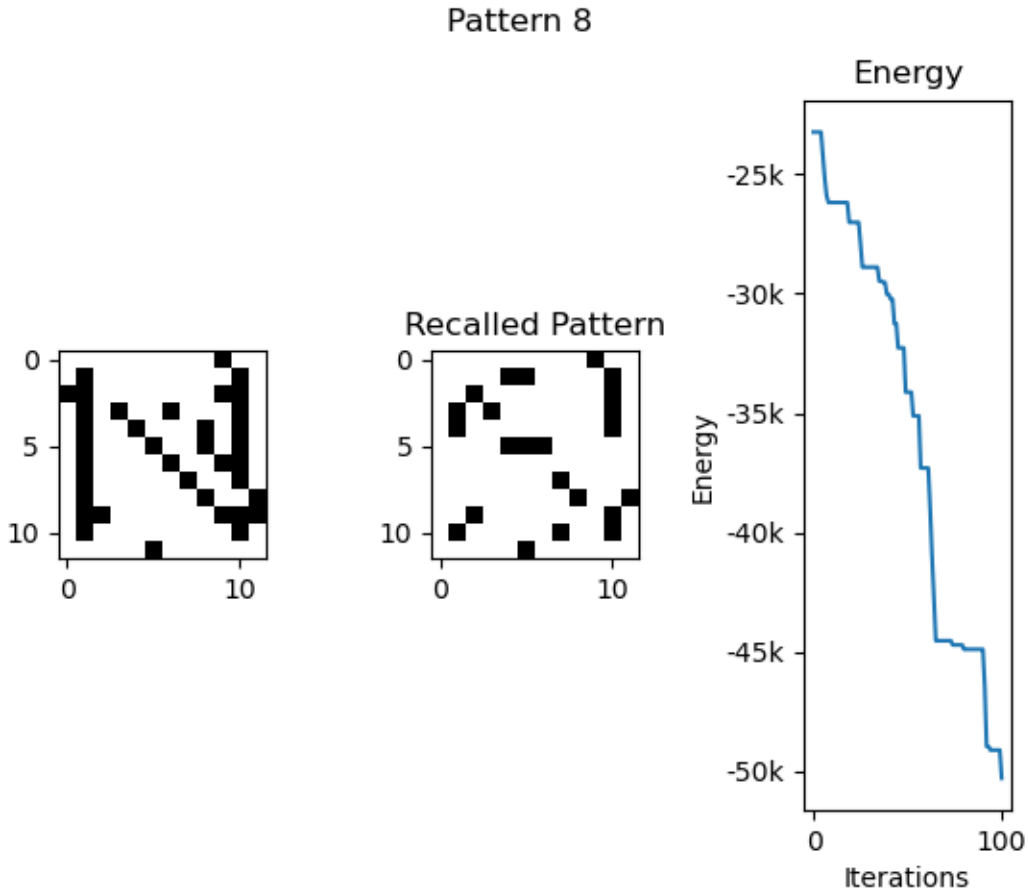

Pattern 5
Recalled Pattern
Energy
0
5
10
0
10
-36k
-37k
-37k
-38k
-38k
-39k
-39k
-40k
-40k
Energy
0
1
Iterations

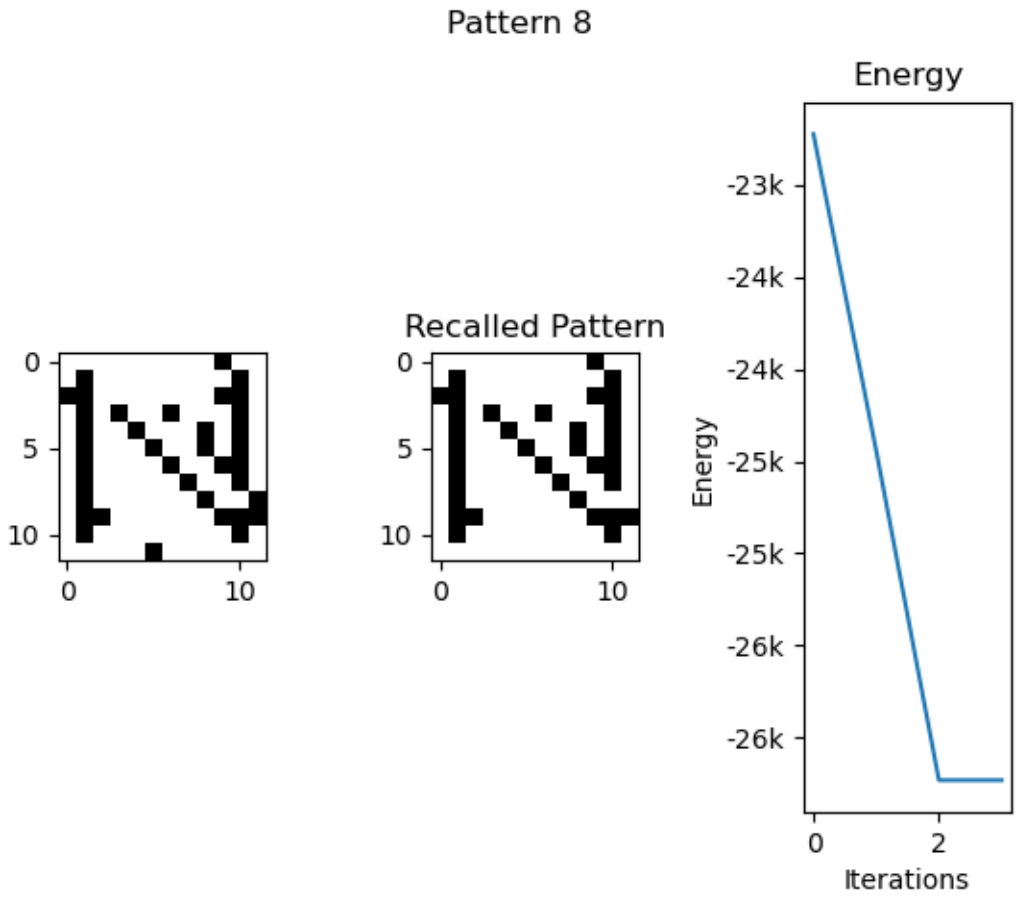

Pattern 6
Recalled Pattern
Energy
0
5
10
0
10
-40k
-45k
-50k
-55k
-60k
-65k
Energy
0
100
Iterations

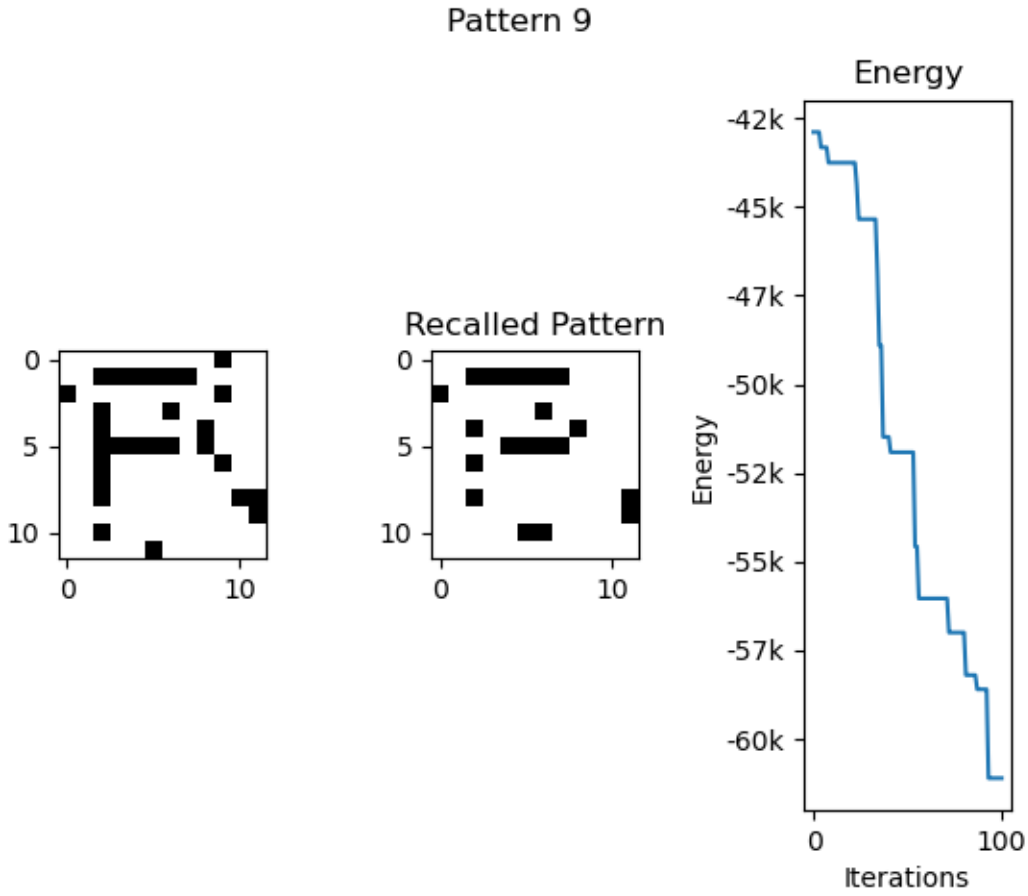

Pattern 6
Recalled Pattern
Energy
0
5
10
0
10
-38k
-39k
-39k
-40k
-40k
-41k
-41k
-42k
Energy
0
1
Iterations

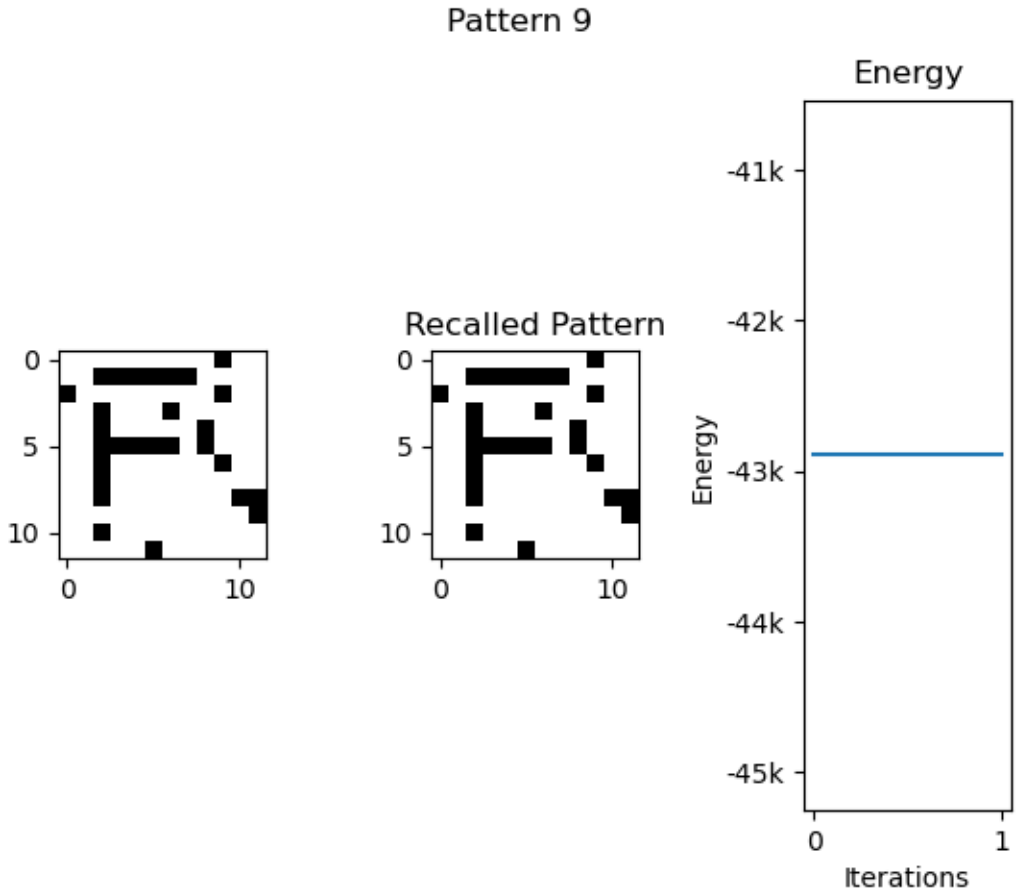

Pattern 7
Recalled Pattern
Energy
0
5
10
0
10
-37k
-40k
-42k
-45k
-47k
-50k
-52k
Energy
0
100
Iterations

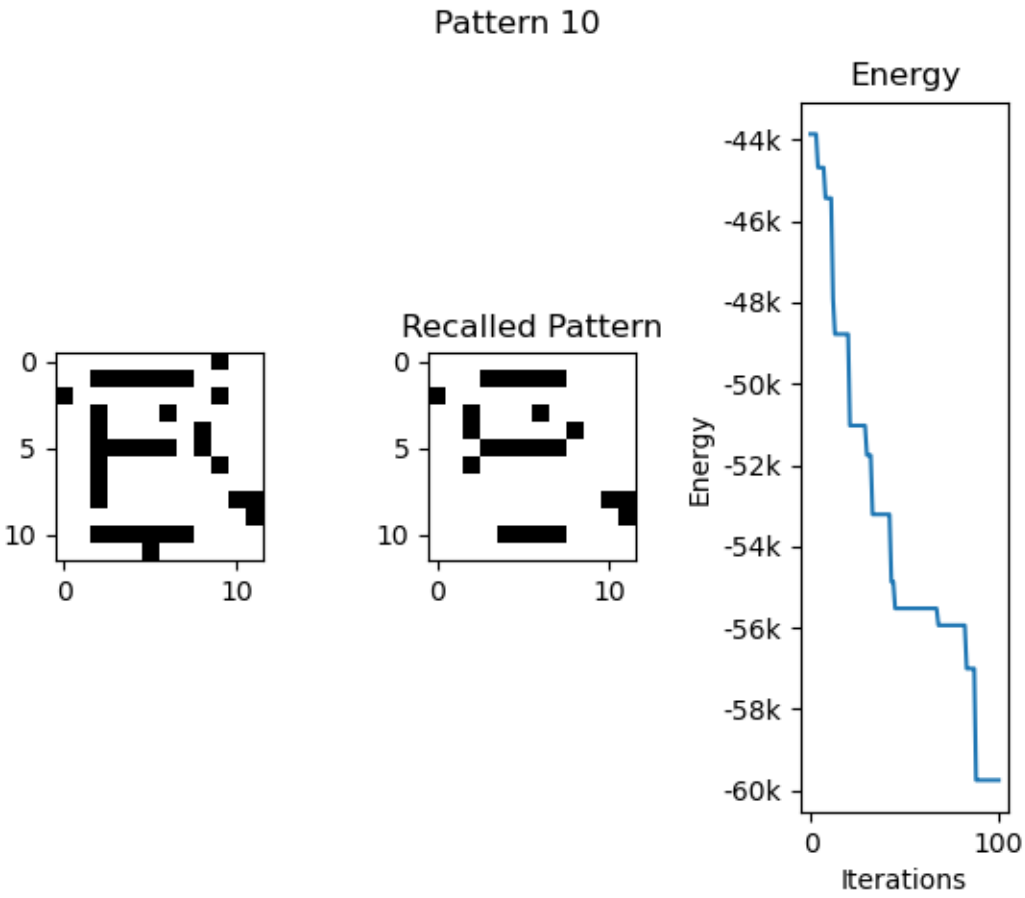

Pattern 7
Recalled Pattern
Energy
0
5
10
0
10
-34k
-35k
-35k
-36k
-36k
-37k
-37k
-38k
Energy
0
1
Iterations

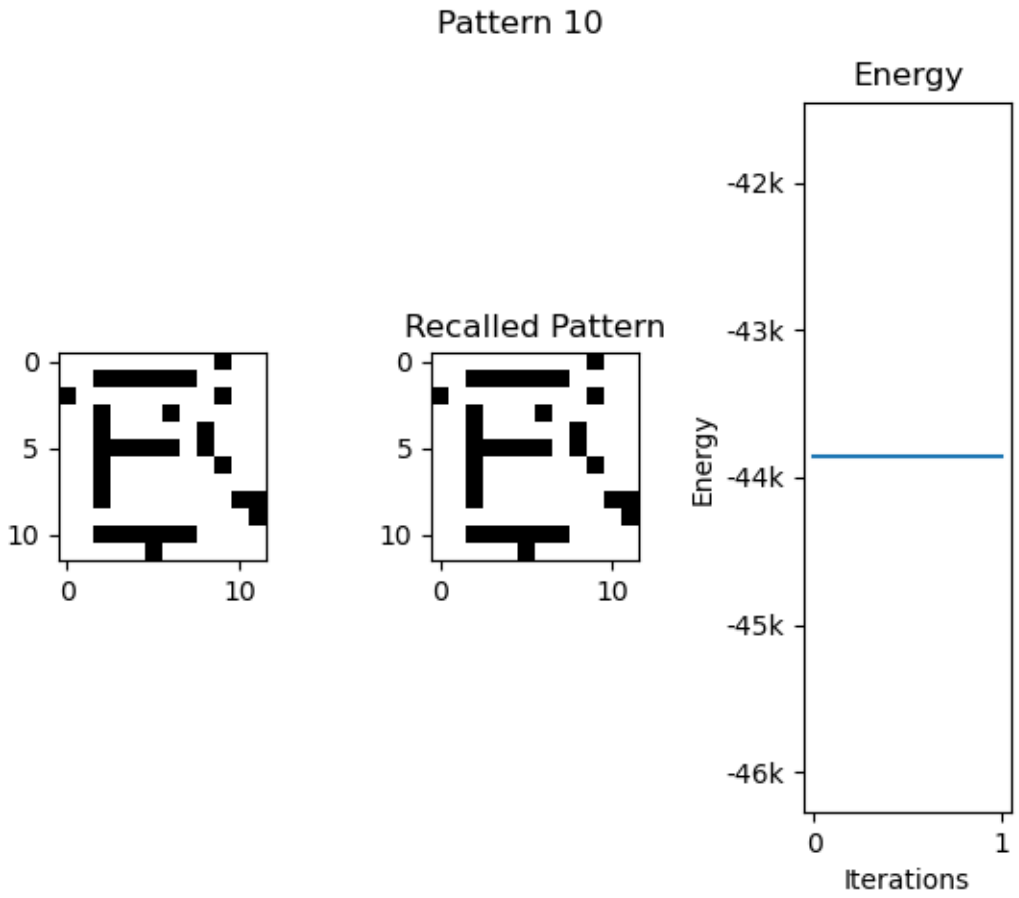

Pattern 8
Recalled Pattern
Energy
0
5
10
0
10
-25k
-30k
-35k
-40k
-45k
-50k
Energy
0
100
Iterations

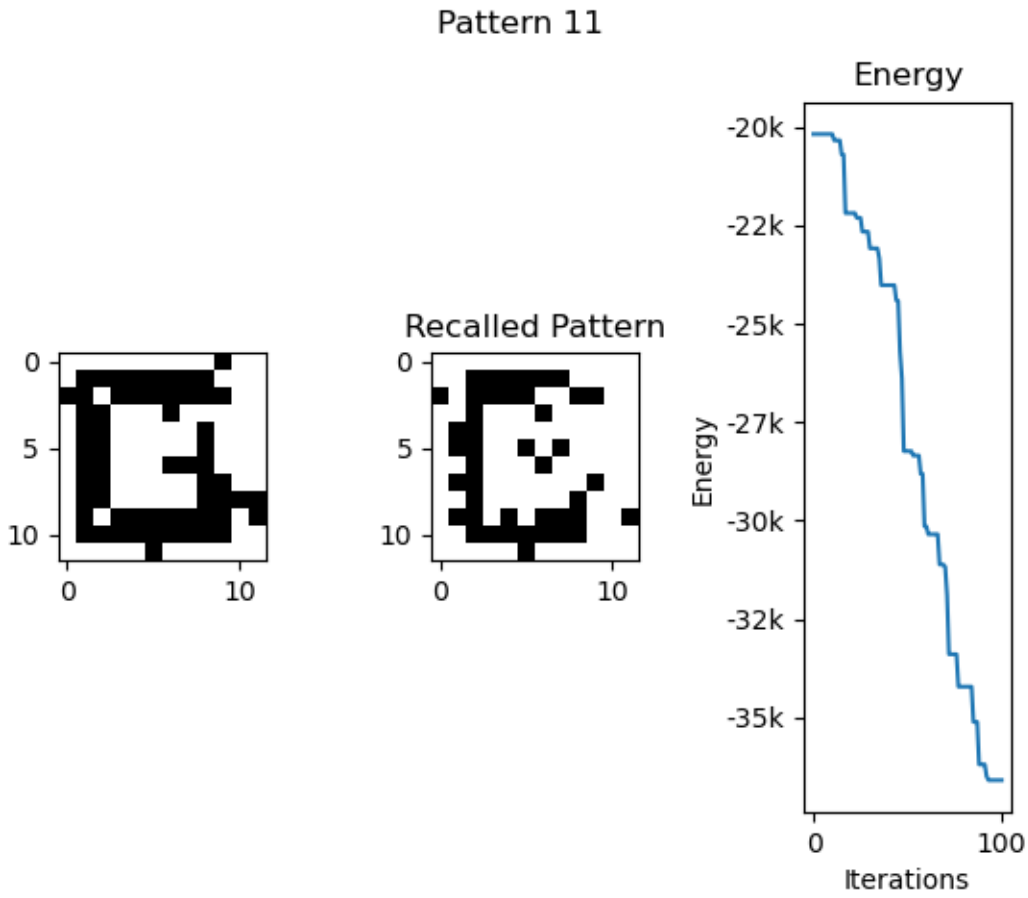

Pattern 8
Recalled Pattern
Energy
0
5
10
0
10
-23k
-24k
-24k
-25k
-25k
-26k
-26k
Energy
0
2
Iterations

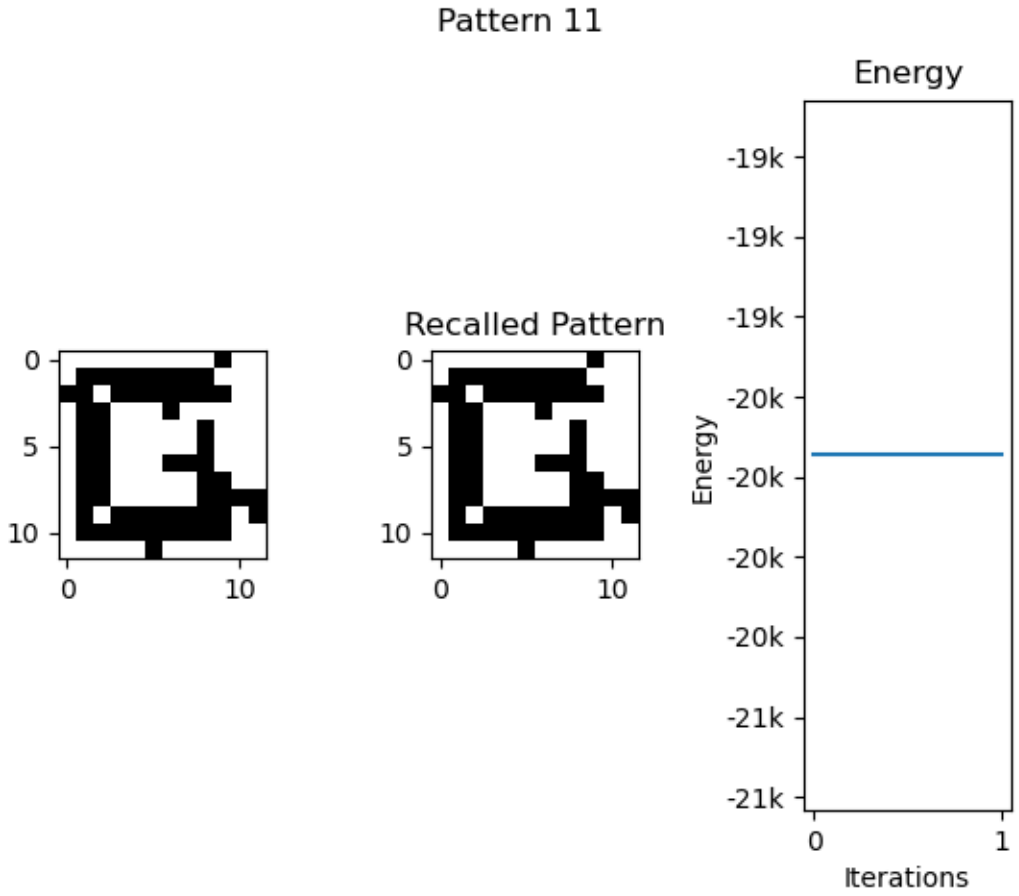

Pattern 9
Recalled Pattern
Energy
0
5
10
0
10
-42k
-45k
-47k
-50k
-52k
-55k
-57k
-60k
Energy
0
100
Iterations

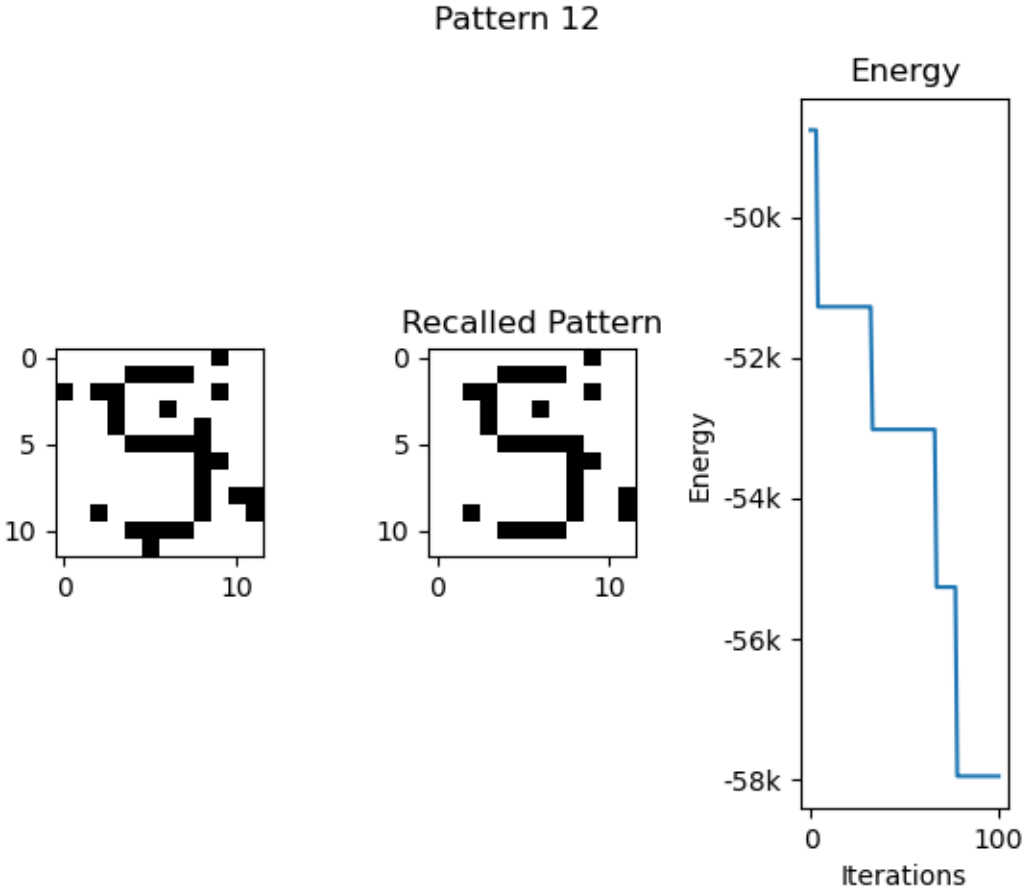

Pattern 9
Recalled Pattern
Energy
0
5
10
0
10
-41k
-42k
-43k
-44k
-45k
Energy
0
1
Iterations

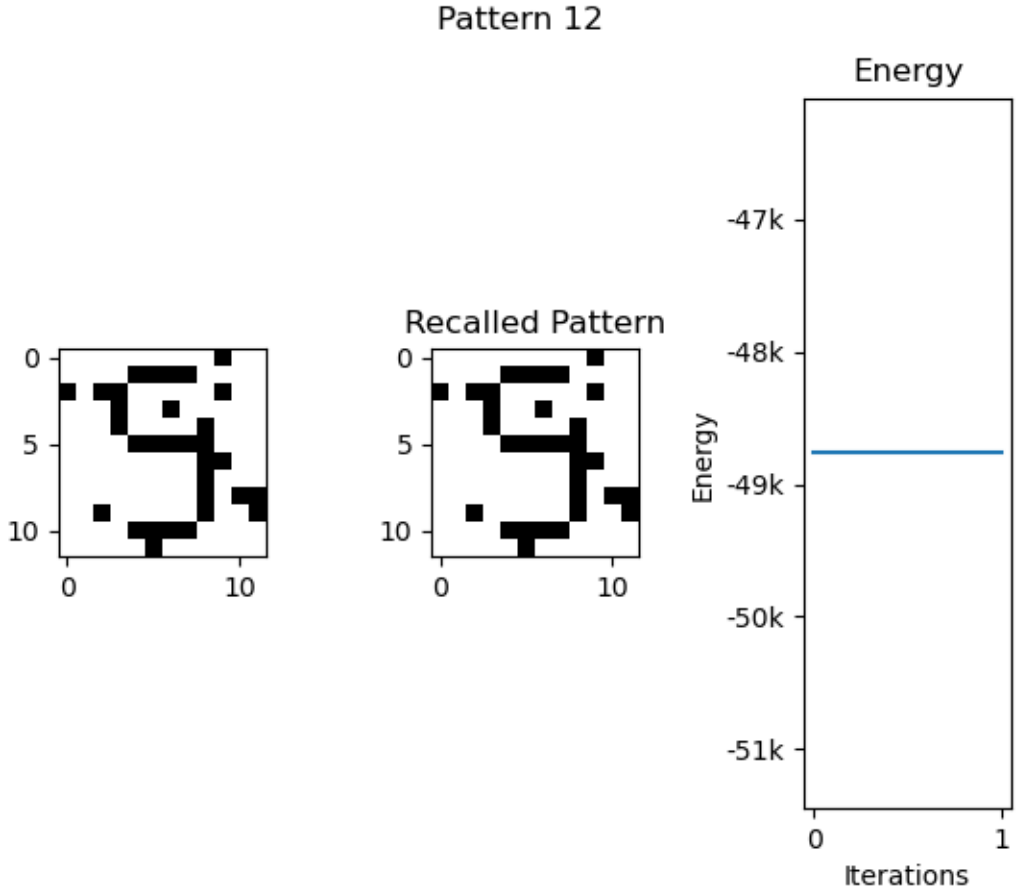

Pattern 10
Recalled Pattern
Energy
0
5
10
0
10
-44k
-46k
-48k
-50k
-52k
-54k
-56k
-58k
-60k
Energy
0
100
Iterations

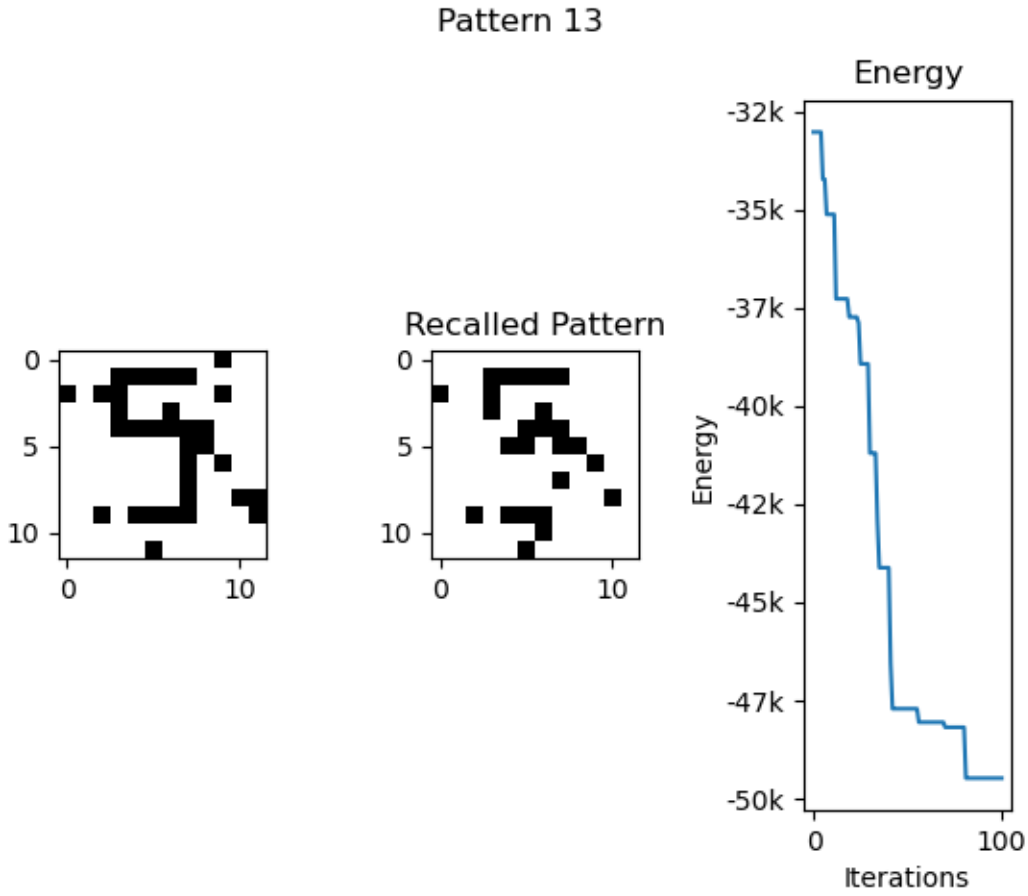

Pattern 10
Recalled Pattern
Energy
0
5
10
0
10
-42k
-43k
-44k
-45k
-46k
Energy
0
1
Iterations

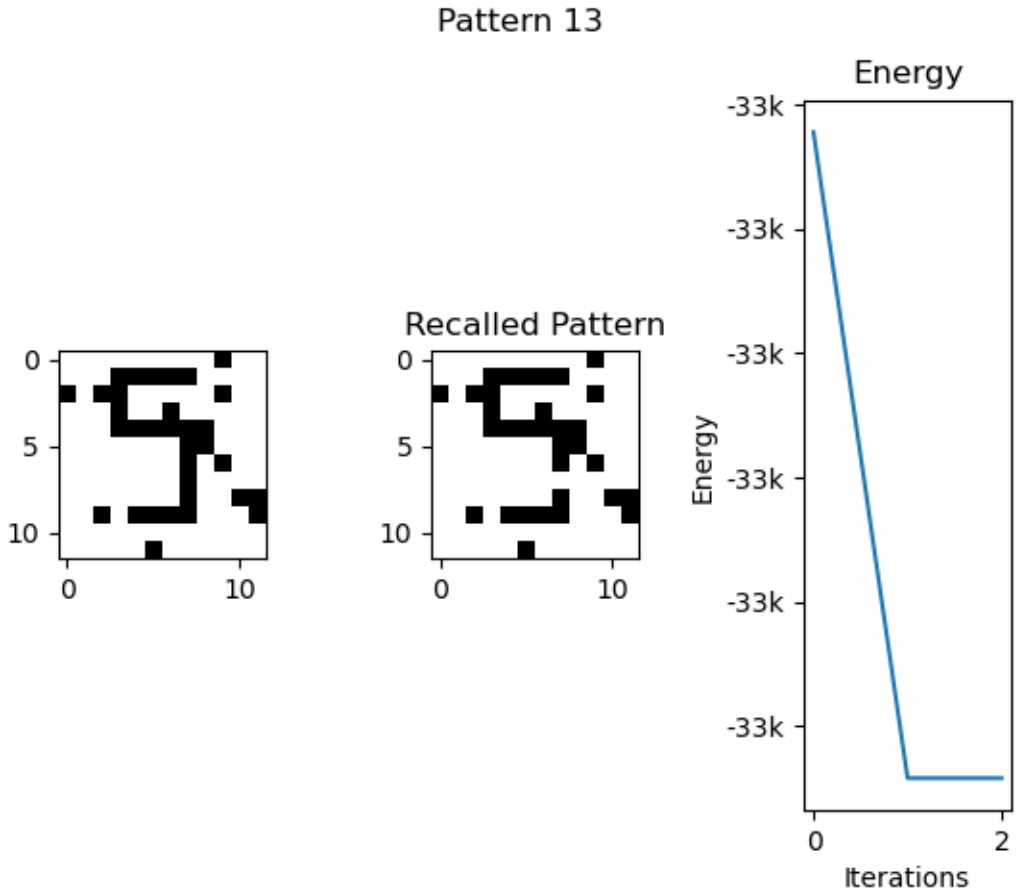

Pattern 11
Recalled Pattern
Energy
0
5
10
0
10
-20k
-22k
-25k
-27k
-30k
-32k
-35k
Energy
0
100
Iterations

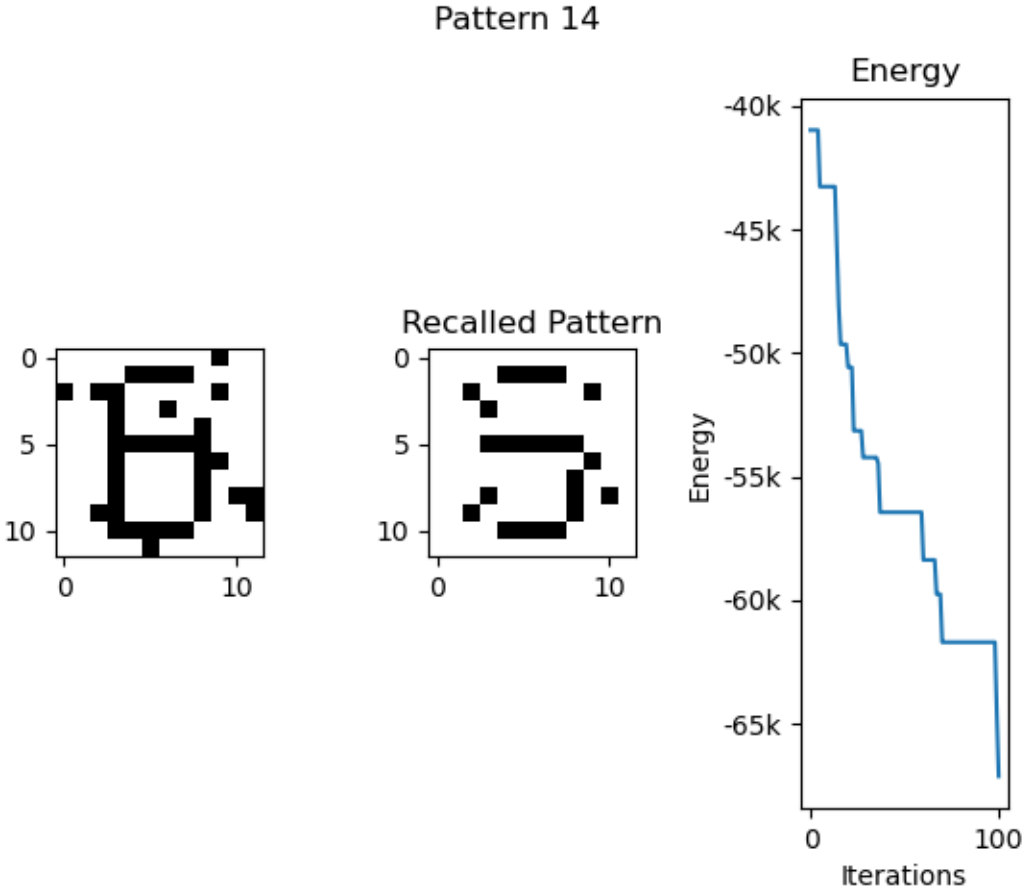

Pattern 11
Recalled Pattern
Energy
0
5
10
0
10
-19k
-19k
-19k
-20k
-20k
-20k
-21k
-21k
Energy
0
1
Iterations

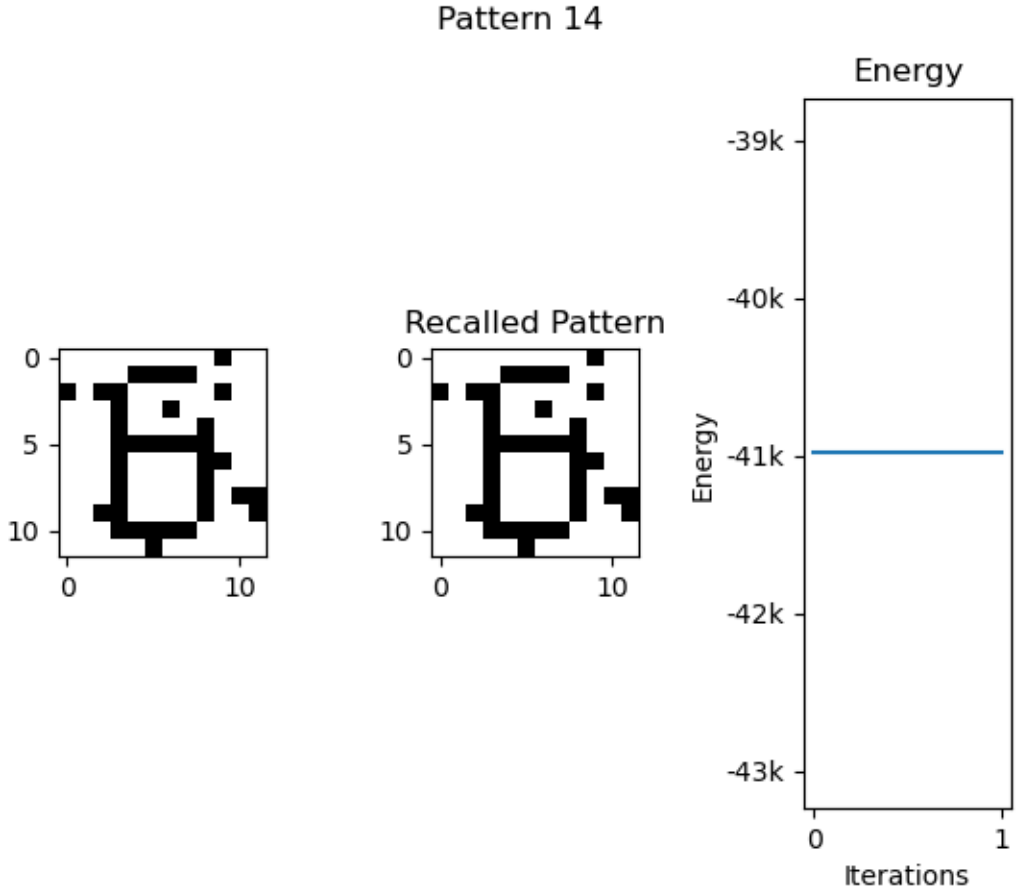

Pattern 12
Recalled Pattern
Energy
0
5
10
0
10
-50k
-52k
-54k
-56k
-58k
Energy
0
100
Iterations

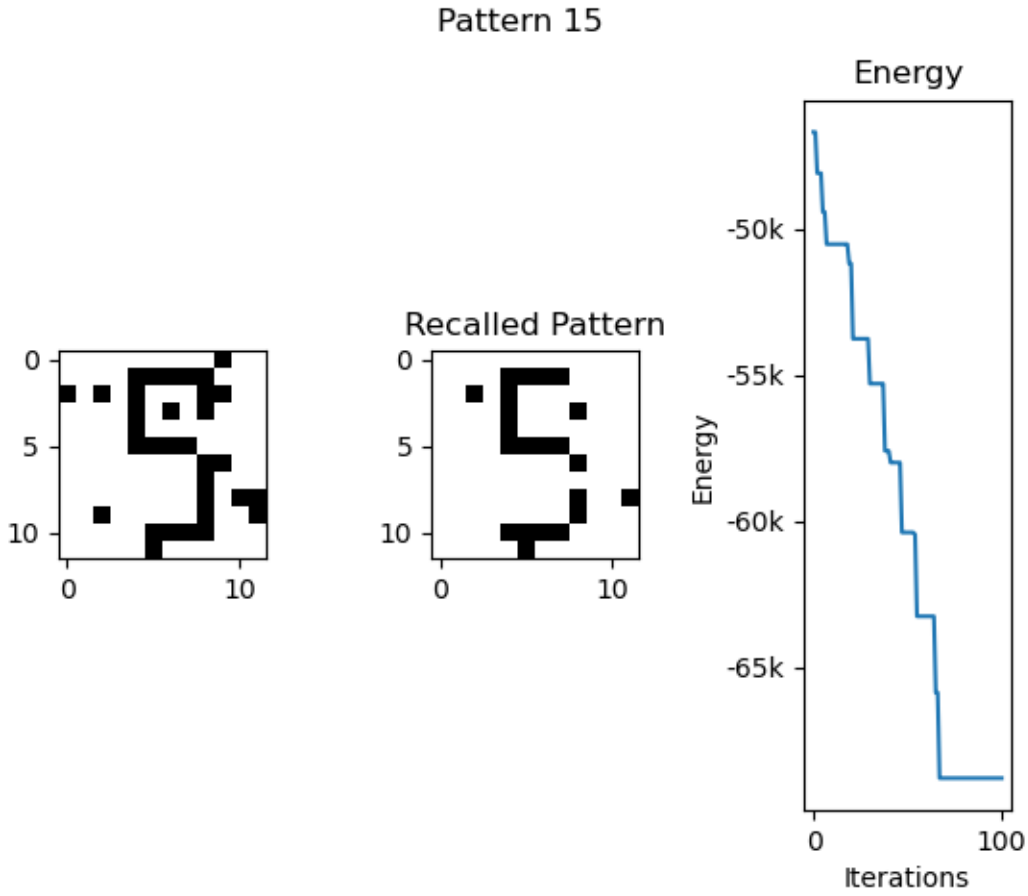

Pattern 12
Recalled Pattern
Energy
0
5
10
0
10
-47k
-48k
-49k
-50k
-51k
Energy
0
1
Iterations

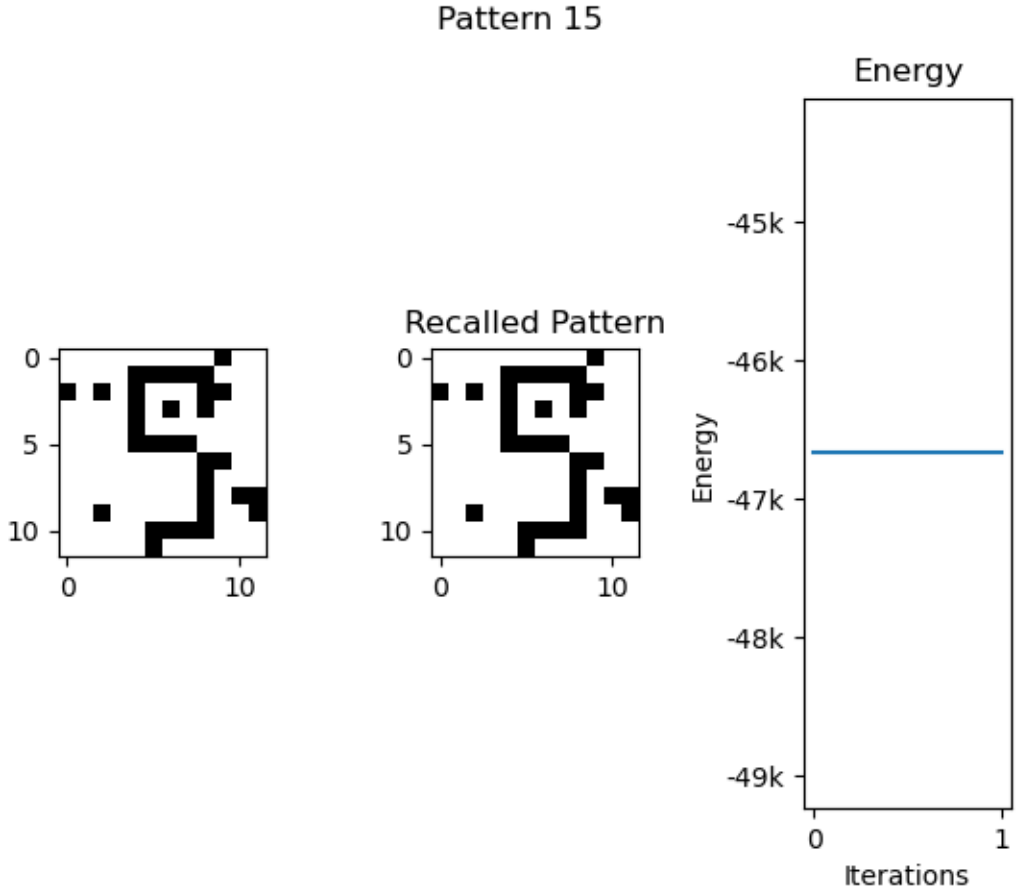

Pattern 13
Recalled Pattern
Energy
0
5
10
0
10
-32k
-35k
-37k
-40k
-42k
-45k
-47k
-50k
0
100
Energy
Iterations

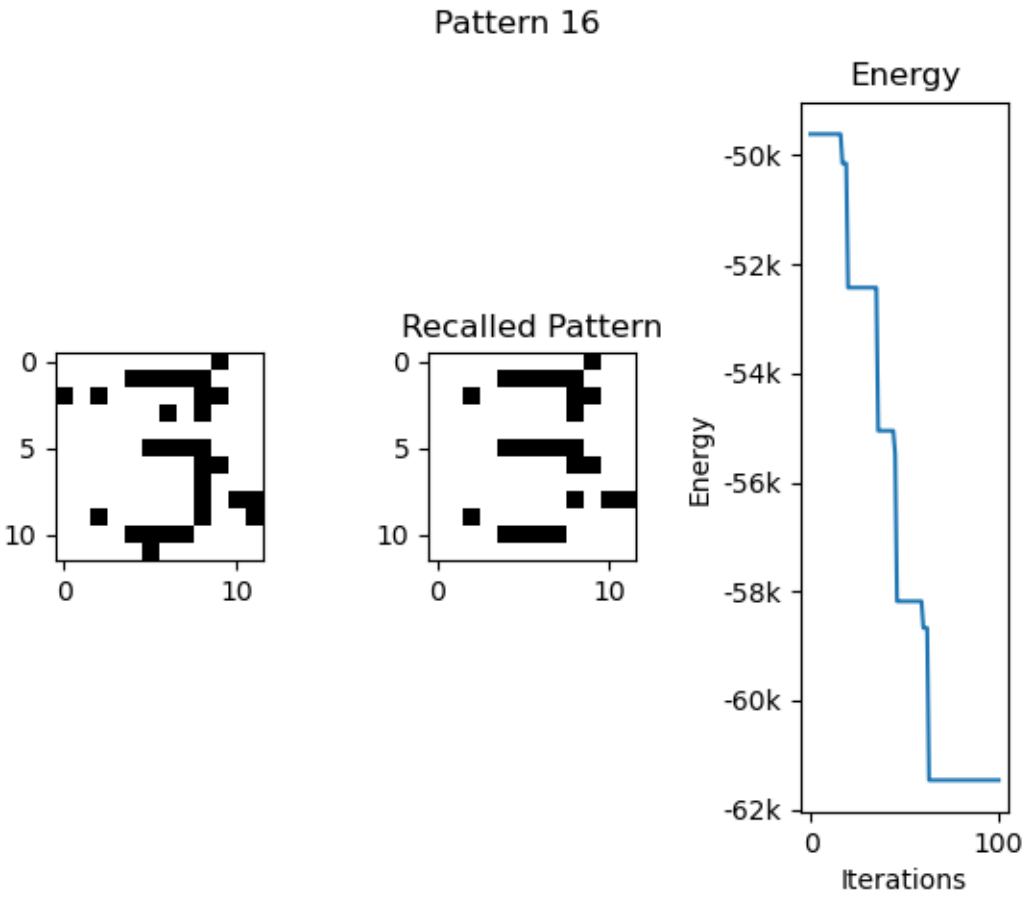

Pattern 13
Recalled Pattern
Energy
0
5
10
0
10
-33k
-33k
-33k
-33k
-33k
-33k
Energy
0
2
Iterations

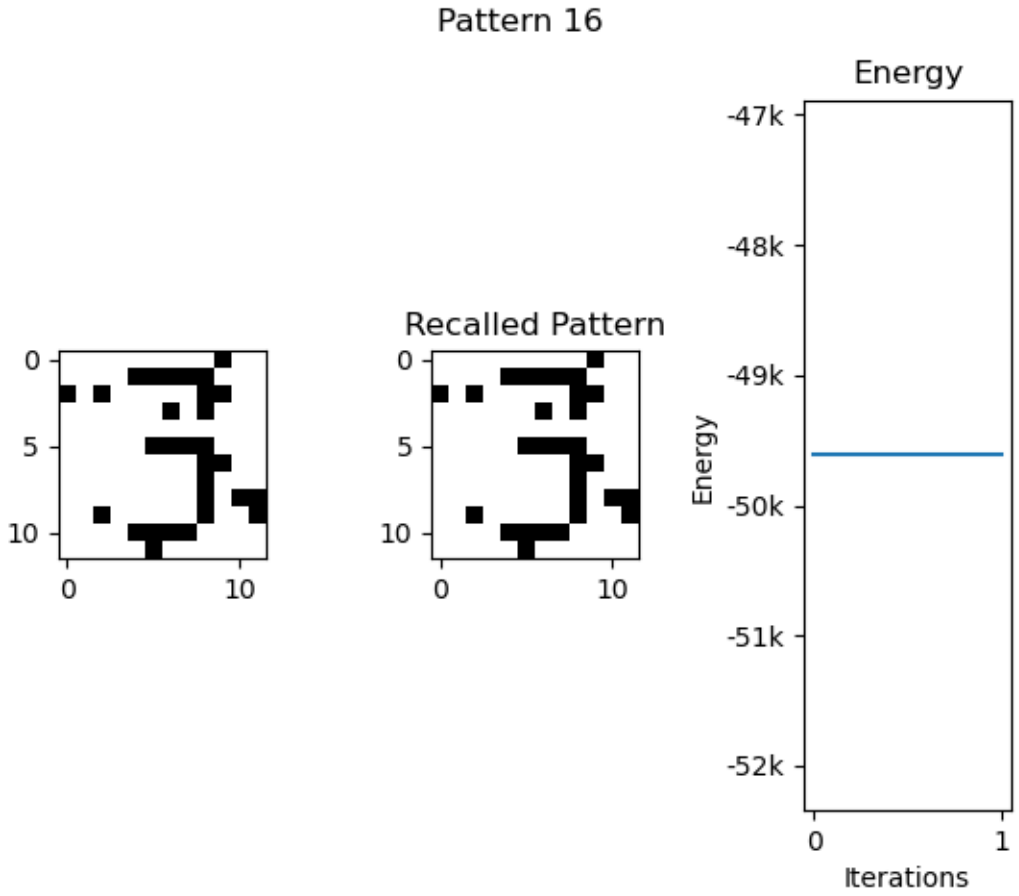

Pattern 14
Recalled Pattern
Energy
0
5
10
0
10
-40k
-45k
-50k
-55k
-60k
-65k
Energy
0
100
Iterations

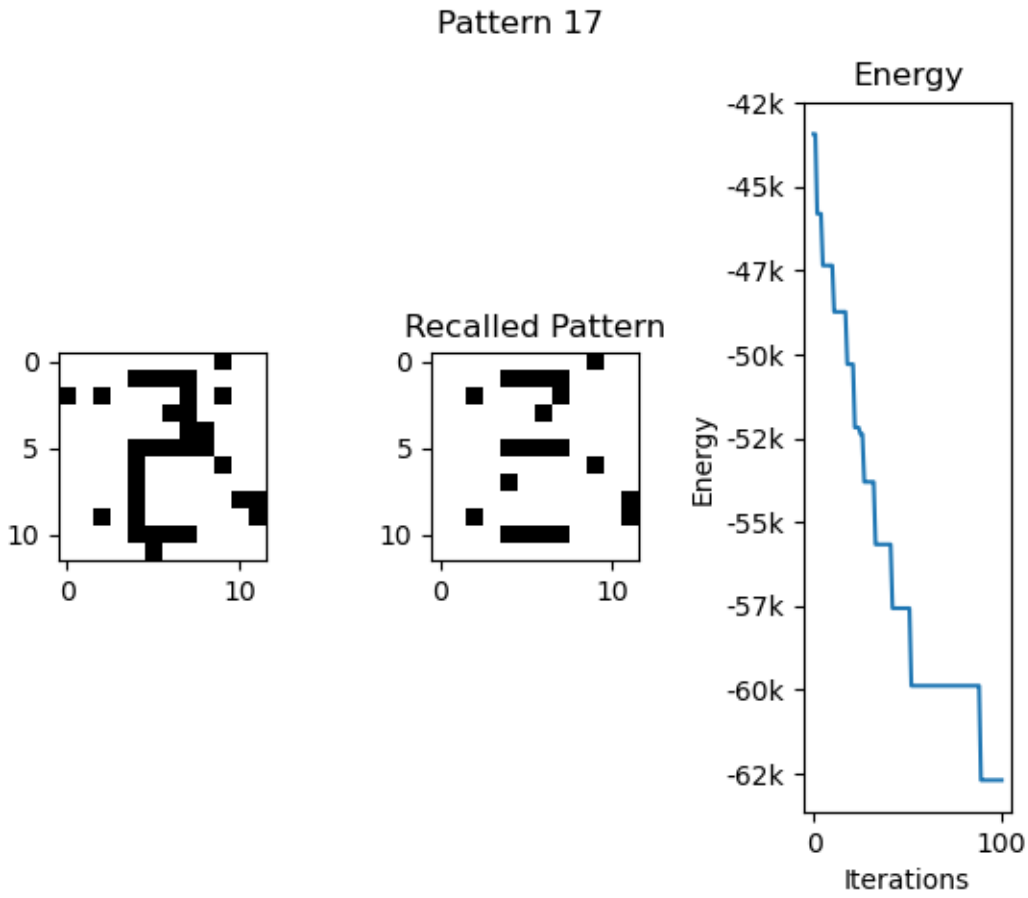

Pattern 14
Recalled Pattern
Energy
-39k
-40k
-41k
-42k
-43k
Energy
0
1
Iterations
0
5
10
0
10

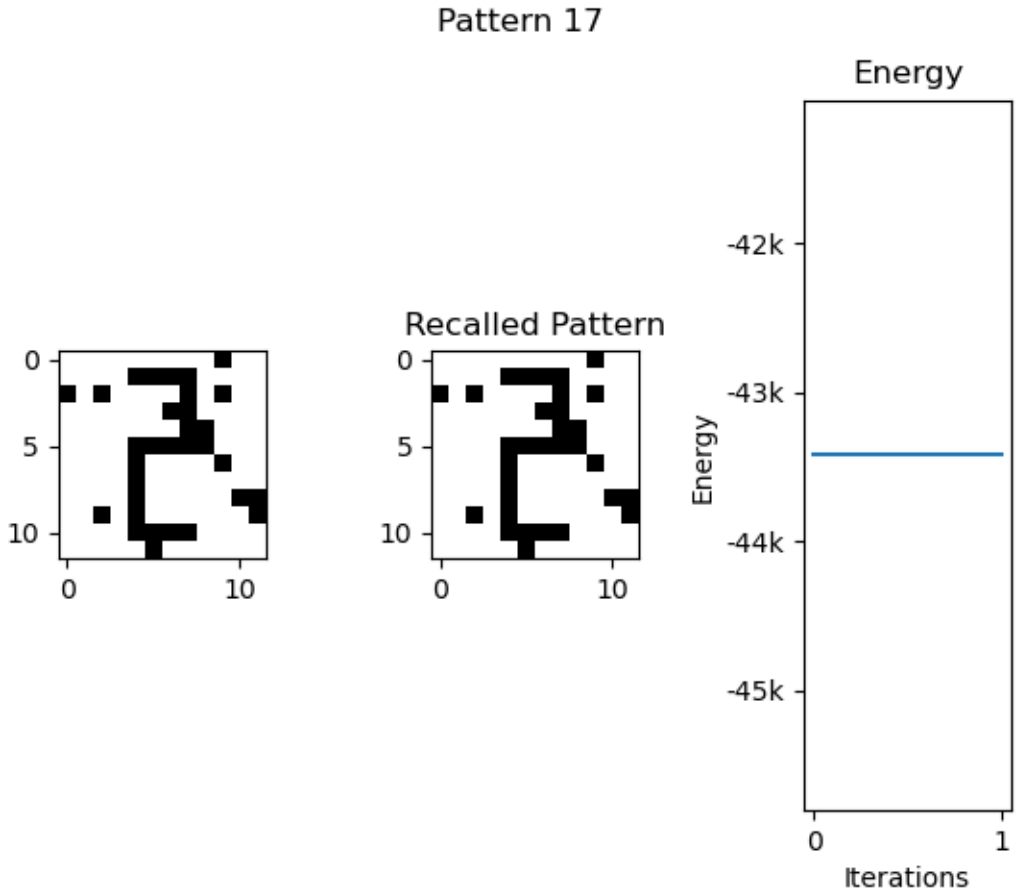

Pattern 15
Recalled Pattern
Energy
0
5
10
0
10
-50k
-55k
-60k
-65k
Energy
0
100
Iterations

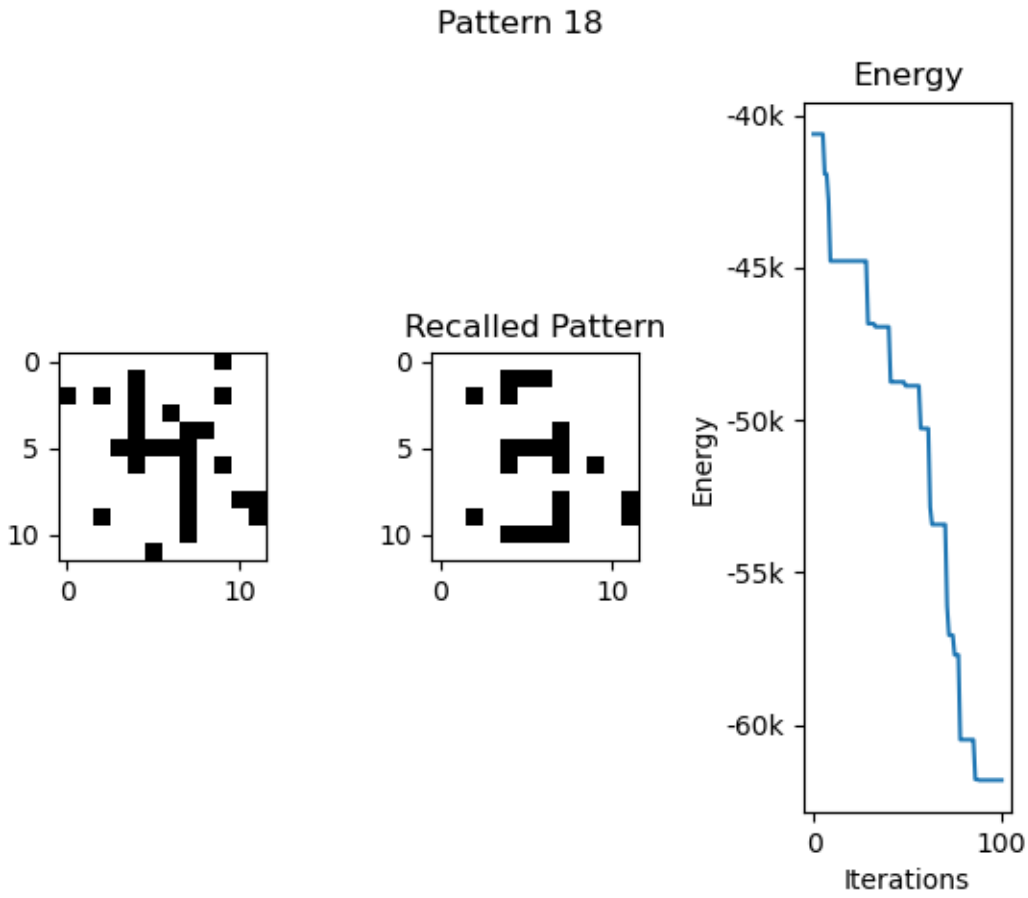

Pattern 15
Recalled Pattern
Energy
0
5
10
0
10
-45k
-46k
-47k
-48k
-49k
Energy
0
1
Iterations

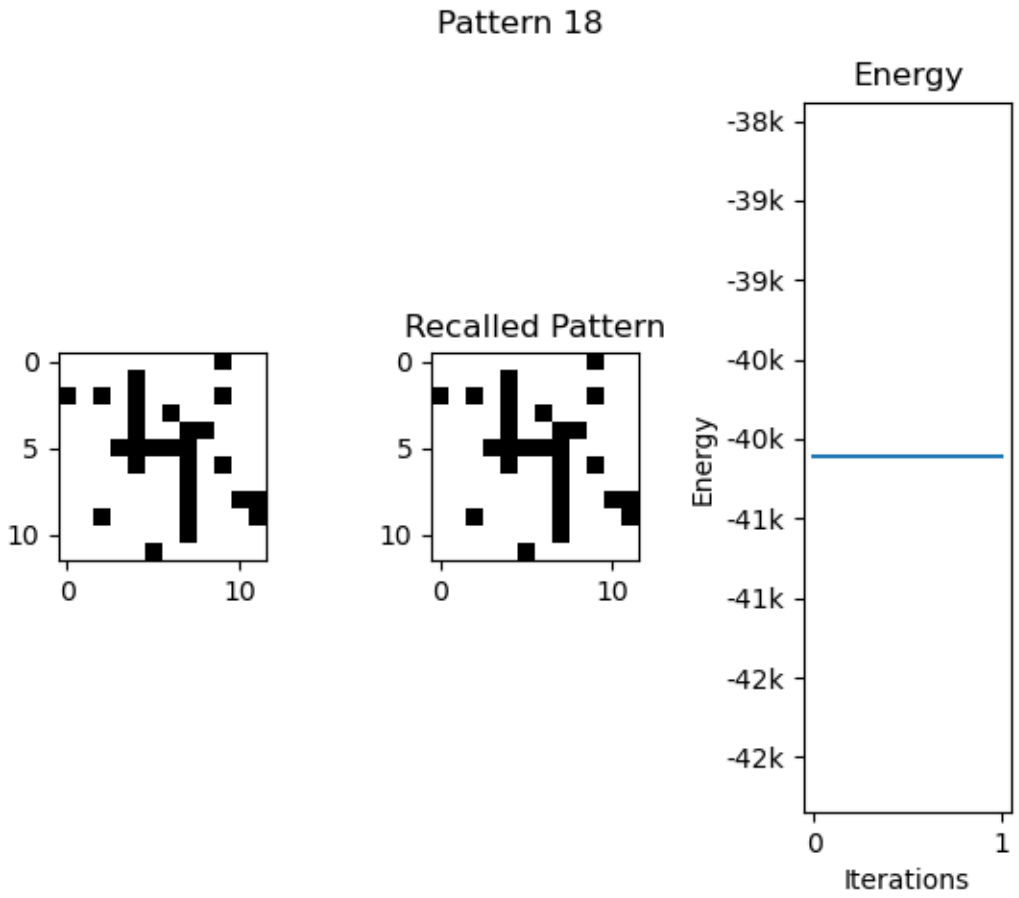

Pattern 16
Recalled Pattern
Energy
0
5
10
0
10
-50k
-52k
-54k
-56k
-58k
-60k
-62k
Energy
0
100
Iterations


Pattern 16
Recalled Pattern
Energy
0
5
10
0
10
-47k
-48k
-49k
-50k
-51k
-52k
Energy
0
1
Iterations


Pattern 17
Recalled Pattern
Energy
0
5
10
0
10
-42k
-45k
-47k
-50k
-52k
-55k
-57k
-60k
-62k
Energy
0
100
Iterations


Pattern 17
Recalled Pattern
Energy
0
5
10
0
10
-42k
-43k
-44k
-45k
Energy
0
1
Iterations


Pattern 18
Recalled Pattern
Energy
0
5
10
0
10
-40k
-45k
-50k
-55k
-60k
Energy
0
100
Iterations



Code:

```
from scipy.io import loadmat
import matplotlib.pyplot as plt
from matplotlib.ticker import FuncFormatter
import numpy as np
import random


def main():
    # indexes = random.sample(range(18), 7)
    train_data = loadmat("characters.mat")
```

```
train_data = train_data["char3"][0]
test_data = loadmat("test.mat")
test_data = test_data["test"][0]

# Flatten the data
patterns = [pattern.reshape(-1, 1).astype(int) for pattern in train_data]
test_patterns = [pattern.reshape(-1, 1).astype(int)
            for pattern in test_data]

for i in range(len(patterns)):
   plt.imshow(patterns[i].reshape(12, 12), cmap="gray")
   plt.savefig(f"input/pattern_{i+1}.png")
   plt.close()

weights = sum([weightMatrix(pattern) for pattern in patterns])

# test "memory" recall
recalled_patterns_and_energies = [updateNetwork(
   pattern, weights) for pattern in patterns]

for i in range(len(patterns)):
   transformationAnalysis(
      i, patterns[i], recalled_patterns_and_energies[i], test=False, early_stop=False)

recalled_patterns_and_energies = [
   updateNetwork(pattern, weights) for pattern in test_patterns
```

```
    ]

    for i in range(len(patterns)):
        transformationAnalysis(
            i, test_patterns[i], recalled_patterns_and_energies[i], test=True, early_stop=False)


def transformationAnalysis(i, pattern, recalled_pattern_and_energies, test=False,
early_stop=False):
    pattern = pattern.copy().reshape(12, 12)
    recalled_pattern, energies = recalled_pattern_and_energies
    recalled_pattern = recalled_pattern.reshape(12, 12)

    fig, ax = plt.subplots(1, 3)
    fig.suptitle(f"Pattern {i+1}")
    ax[0].imshow(pattern, cmap="gray")
    ax[1].imshow(recalled_pattern, cmap="gray")
    ax[1].set_title("Recalled Pattern")
    ax[2].plot(energies)
    ax[2].set_title("Energy")
    ax[2].set_xlabel("Iterations")
    ax[2].set_ylabel("Energy")
    ax[2].yaxis.set_major_formatter(
        FuncFormatter(lambda x, _: f"{int(x/1000)}k" if abs(x) > 1000 else x)
    )
    plt.subplots_adjust(
```

```
    left=None, bottom=None, right=None, top=None, wspace=0.8, hspace=None
  )

  filename = f"test_{i+1}" if test else f"pattern_{i+1}"
  filename += "_early_stop" if early_stop else ""
  plt.savefig(f"results/{filename}.png")
  plt.close()

  # TODO calculate error between pattern and recalled pattern, display calculated error
  print(
    f"Error between pattern {i+1} and recalled pattern: {np.sum(np.abs(pattern - recalled_pattern))}")



def updateNetwork(pattern, weights, max_iterations=100):

  pattern = (pattern * 2) - 1
  energies = [Energy(pattern, weights)]

  for _ in range(max_iterations):
    # for i in random.sample(range(len(pattern)), len(pattern)):
    i = random.randint(0, len(pattern) - 1)
    weighted_sum = np.dot(weights[i, :], pattern)
    pattern[i] = 1 if weighted_sum > 0 else -1

    current_energy = Energy(pattern, weights)
```

```
        energies.append(current_energy)

        # if energies[-1] == energies[-2]:
        #     break

    return (pattern + 1) // 2, energies


def Energy(pattern, weights):
    energy = 0
    for i in range(len(pattern)):
        for j in range(len(pattern)):
            energy += weights[i][j] * pattern[i] * pattern[j]

    return energy / (-2)


def weightMatrix(query):
    # Create a 2D array of the same size as the query
    query = query * 2 - 1
    w_matrix = np.outer(query, query)
    np.fill_diagonal(w_matrix, 0)
    return w_matrix
```

```
if __name__ == "__main__":
    main()
```