```
 1: def iterate(self):
 2:     import numpy as np
 3:     self._x_value = self._start
 4:     old_x_value = None
 5:     self._iteration = 0
 6:     self._sum = np.zeros(self._x_value.shape)
 7:     alpha_n = np.zeros(self._x_value.shape)
 8:     alpha_n.fill(self._step_size)
 9:     self._converged_value = False
10:     self._grad_value = self._gradient(self._x_value)
11:     self._step_coeff = self._step_size
12:
13:     yield self.state_dict()
14:
15:     while not self._converged_value:
16:         self._iteration += 1
17:         if self._max_iter > 0 and self._iteration > self._max_iter:
18:             break
19:         self._grad_value = self._gradient(self._x_value)
20:         old_x_value = self._x_value
21:         self._x_value = self._x_value - alpha_n * self._grad_value
22:         self._sum = self._beta * self._sum + (1-self._beta) * (self._grad_value**2)
23:         alpha_n = self._step_size / (self._sum**0.5+self._epsilon)
24:         self._step_coeff = alpha_n
25:         self._converged_value = self._converged(self._x_value, old_x_value)
26:         yield self.state_dict()
```