```python
from particle_swarm_optimization import Particle
from particle_swarm_optimization import ParticleSwarmOptimization
from backpropagation_neural_net import BackpropagationNN, Neuron
from random import random
# import pandas as pd
# from matplotlib import pyplot as plt


class BackpropagationPSO(BackpropagationNN):

    def __init__(self, input, hidden, output, learning_rate):
        super().__init__(input, hidden, output, learning_rate)

    def initWeight(self, partikel):
        layer = list()
        partikel_dimens_idx = 0
        input_to_hidden = list()

        for i in range(self.HIDDEN_LAYER):
            w = list()
            for j in range(self.INPUT_LAYER):
                w.append(partikel[partikel_dimens_idx])
                partikel_dimens_idx += 1

            w.append(random())
            input_to_hidden.append(Neuron(w))
        layer.append(input_to_hidden)

        hidden_to_output = list()
        for i in range(self.OUTPUT_LAYER):
            w = list()
            for j in range(self.HIDDEN_LAYER + 1):
                w.append(random())
            hidden_to_output.append(Neuron(w))
```