Nama : Fikri Anun Nafih

NIM : 2004111 00153

Kelas : KKE

Latihan

① Ekivalensi dari bimbikasi adalah $Y = (X \rightarrow Y)$ dan $(Y \leftarrow X)$

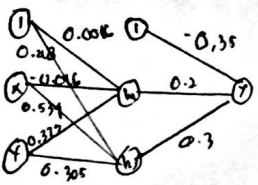
Oleh karena itu lanjutan pembelajaran terlebih dahulu

Jika $X \rightarrow Y$ = bias = 0.0046, $w_1 = -0.046$ dan $w_2 = 0.373$

dan $Y \leftarrow X$ = bias = 0.218, $w_1 = 0.534$, $w_3 = 0.305$

hasil 14 = bias = -0.35, $w_1 = 0.2$ dan $w_3 = 0.3$

arsitektur pendekatan



② kode untuk soal latihan 2

```
CV = np.array([[ 0.9 ,0.7, 0.1, 0.2],[ 0.2 , 0.2, 0.7 ,0.9]])
data : np.array([[ 1,1,1,0],[ 0,0,0,1] ,[ 1,0,0,0],[ 0.0, 1,1]])
jumlah epoch = 1
learning rate : 0.5
for k in range ( jumlahepoch ):
for i in range ( len(data)):
tempDish = np. zeros (len(cv)):
tempDish[j] = np. linalg . norm (data [i] - cv[j]):
minind = np. argmin (tempdish)
for j in range (len (cv)):
cv[j] = cv[j] + learningrate * neighborhood fn ( cv[ min ind ], cv[j])
Print (cv)
For i in range ( len (data1)):
tempDish = np. zeros ( len(cv))
for j in range ( len (cv))
tempDish [j] = np. linalg. norm (data [i] - cv[j])
minind = np. argmin (tempDish)
Print (cvnid).
```

Nama : Fitri Ainun Najib
NIM : 2009 1100153
Kelas : KKE

Resume

* Particle swarm optimization

merupakan bagian dari pendekatan swarm inteligence yang terinspirasi dari pola terbang dari sekelompok burung yang berpindah tempat dari suatu daerah ke daerah lainnya. Dari pergerakan koloni tersebut dianggap sebagai agen untuk mencari solusi dalam ruang pencarian. Oleh karena itu PSO ini digunakan untuk menjadi permasalahan optimasi.

masing - masing partikel yang dibangkitkan secara acak, bergerak untuk menemukan solusi terbaik. pergerakan partikel - partikel ini dipengaruhi oleh:

1. kecepatan partikel (inertia velocity).
2. selisih atau jarak antara posisi partikel saat ini (current position). dengan personal best position dari partikel itu sendiri ..
3. selisih atau jarak antara posisi partikel saat ini dengan global best position.

misal, $x_i(t)$ merupakan partikel ke-i pada waktu ke-t maka partikel akan diperbaharui melalui persamaan:

$x_i(t+1) = x_i(t) + v_i(t+1)$.

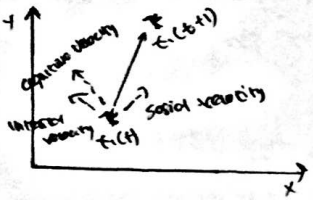
Sedangkan v_i akan dihitung melalui persamaan.

$v_i(t+1) = c_0 v_i(t) + c_1 r_1(t)[p_i(t) - x_i(t)] + c_2 r_2(t)[p_g(t) - x_i(t)]$.

Dimana c_1 dan c_2 merupakan konstanta percepatan dari cognitive. r_1 dan r_2 merupakan element stokastik yg dibangkitkan random.

Berikut algoritma PSO

1. Bangkitkan sejumlah n partikel (beserta kecepatan dan kecepatan) secara random.
2. Selama kondisi terpenuhi maka lakukan perulangan berikut
   a. Untuk setiap partikel ke-i :
      1) $X_i < f(P_i)$ maka $P_i = X_i$
      2) Min (neighbourd)
      3) Update Velocity
      4) Update partikel.



Ant colony Optimization

Merupakan terinspirasi dari kehidupan koloni semut khususnya pada kerjasama antara semut dalam pencarian sumber makanan. Oleh karena itu algoritma ACO terinspirasi dari perilaku unik dari koloni semut ini, yang ternyata dapat menghasilkan hasil optimal dengan bantuan pheromone dan kecerdasan individu semut.

Terdapat dua tahapan utama dalam algoritma ACO :

1. Pembentukan kandidat solusi
2. Update konsentrasi pheromones.

- Pembentukan kandidat solusi

Probabilitas rute ini dipengaruhi oleh informasi prior dan posterior kedekatan masing-masing path menggunakan persamaan :

$$P_{ij}^{k}(t) = \frac{T_{ij}^{\alpha} \eta_{ij}^{\beta}}{\sum_{l \in N_i(t)} T_{il}^{\alpha} \eta_{il}^{\beta}} \quad \text{dan} \quad \eta_{ij} = \frac{1}{d_{ij}}$$

k = semut ke-n

T_{ij} = informasi posterior

η_{ij} = informasi prior

- Update konsentrasi pheromone

Semakin jauh jumlah jarak maka pheromone akan semakin cepat menguap. evaporasi pheromone dapat dihitung di persamaan berikut:

$$\tau_{ij}(t) = (1-\rho)\,\tau_{ij}(t).$$

Untuk update nilai pheromone, terlebih dahulu menghitung nilai delta.

$$\Delta_{ij}^{k}(t) = \frac{Q}{f(x^{k}(t))} \text{ jika } (i,j) \text{ terdapat di rute } x^{k}(t)$$

$$= 0 \text{ jika selainnya.}$$

Sedangkan update pheromone dapat dihitung dengan persamaan berikut

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \sum_{k=1}^{nk} \Delta_{ij}^{k}(t)$$

Algoritma ACO ini digunakan untuk memecahkan permasalahan optimasi seperti PSO. Jika pso di presentasikan dengan partikel dan ACO di representasikan dengan semut.