**FORWARD-ONLY COUNTERPROPAGATION NETWORK:**

**Step 0:** Initialize the weights and learning rates.

**Step 1:** Perform step 2 to 7 when stopping condition for phase I training is false.

**Step 2:** Perform step 3 to 5 for each of training  input X.

**Step 3:** Set the X-input layer activation to vector X.

**Step 4:** Compute the winning cluster unit J. If dot product method is used, find the cluster unit zJ with the largest net input:

zinj=∑xi.vij

If Euclidean distance is used, find the cluster unit zJ square of whose distance from the input pattern is smallest:

Dj=∑(xi-vij)^2

If there exists a tie in the selection of winner unit, the unit with the smallest index is chosen as the winner.

**Step 5:** Perform weight updation for unit zJ. For i=1 to n,

viJ(new)=viJ(old) + α[xi-viJ(old)]

**Step 6:** Reduce learning rate α:

α (t+1)=0.5α(t)

**Step 7:** Test the stopping condition for phase I training.

**Step 8:** Perform step 9 to 1 when stopping condition for phase II training is false.

**Step 9:** Perform step 10 to 13 for each training input pair x:y.

**Step 10:** Set X-input layer activations to vector X. Set Y-output layer activation to vector Y.

**Step 11:** Find the winning cluster unit J.

**Step 12:** Update the weights into unit zJ. For i=1 to n,

viJ(new)=viJ(old) + α[xi-viJ(old)]

**Step 13:** Update the weights from unit zJ to the output units.

For k=1 to m,   wJk(new)=wJk(old) + β[yk-wJk(old)]

**Step 14:** Reduce learning rate β,

β(t+1)=0.5β(t)

**Step 15:** Test the stopping condition for phase II training.