Sheet 4

November 15, 2015

Assignment 19

Assignment 20

Given an array of patterns `P` with `n` elements and a function called `random` useful to calculate integer random number between a range it is possible to shuffle the original array using the next algorithm:

Randomize elements in an array

---

for $i = 0$ `such that` $i < n$ **do**
  $i \leftarrow i + 1$
  $swap \leftarrow random(0, n)$
  $temp \leftarrow N[swap]$
  $N[swap] \leftarrow N[i]$
  $N[i] \leftarrow temp$
end for

Assignment 21

Backpropagation of errors (BP). After the computation of the output values, we calculate the difference between our teacher value ${}^{p}\hat{y}_m$ and each output unit ${}^{p}y_m$. We obtain the sum of all these differences and apply to it an error function:

$$ {}^{p}E = \frac{1}{2} \sum_{m=1}^{M} ({}^{p}\hat{y}_m - {}^{p}y_m)^2 $$

Now, with a derivation of the error function (ommited here) is possible to get a new formula to calculate δ *values*. This calculation is different for the neurons located in the output layer:

$$ \delta_m = (\hat{y}_m - y_m) \cdot f'(net_m) $$

and the neurons located in the hidden layers:

$$\delta_h = (\sum_{k=1}^{k} \delta_k w_{hk}) \cdot f'(net_h)$$

We must be warned that δ_k and w_{hk} are referring to values in the next layer. Now, using the δ *value*, the output $\widetilde{out}_g$, and a learning rate η we can calculate the weight changes is using this formula:

$$\Delta w_{ij} = \eta \cdot \delta w_{ij} \cdot \widetilde{out}_g$$

We iterate in this process until we reach the input layer. Finally, we update the current weights w_{ij} adding to them the changed weights Δw_{ij}:

$$w_{ij} = w_{ij} + \Delta w_{ij}$$

Assignment 22

Assignment 23