NN Implementation

June 26, 2017

Run Step

- $o_{k,j}$: output of layer k, neuron j
- max: n_layers
- $w_{k,j,i}$: weights of layer k, neuron j and weight i
- d_i: i-th data
- r_i: i-th result
- n_l: Number of neurons of layer l

$$o_{1,j} = \sigma(\sum_{i=0}^{n_0} w_{1,j,i} d_i)$$

$$2 <= k < n_{max}, o_{k,j} = \sigma(\sum_{i=0}^{n_{k-1}} w_{k,j,i} o_{k-1,i})$$

$$r_j = \sigma(\sum w_{max,j,i} w_{max,j,i} o_{max-1,i})$$

Train Step

- e_i: i-th expectation
- eta: learning rate
- m=MSE

For last layer:

$$run(d, r)$$

$$\Delta_{max,i} = e_i - r_i$$

$$m = \sum_{i=0}^{max} \mathbf{\Delta^2}_{max,i}$$

$$\Delta_{max,i} *= r_i * (1 - r_i)$$

For other layers:

$$max - 1 > k > 1, \Delta_{k-1,j} = (\sum_{i=0}^{n_k} \Delta_{k,i} * w_{k,i,j}) * o_{k-1,j} * (1 - o_{k-1,j})$$

Update hidden layers weights:

$$max - 1 > k > 1, w_{k,i,j} += eta * \Delta_{k,j} o_{k-1,i}$$

$$w_{1,j,i} += eta * \Delta_{1,j} * d_i$$