Parallel fully connected implementations

Let N = the number of input features,
M = the number of hidden neurons (in our case it is always 40),
C = the number of output neurons/ number of classes
x_i be the i_th_ input feature,
h_i be the i_th_ hidden neuron, also used to denote it's output value before binarization,
s_i be the i_th_ hidden neuron's output after binarization, so $s_i = h_i \geq 0$,
y_i be the i_th output neuron, also used to denote it's output value,
$W1$ = the weight matrix of the first layer,
$W2$ = the weight matrix of the second layer,
rows represent neurons and columns represent input activations, so $W1_{i,j}$ is the weight of the first layer that corresponds to the connection between the input feature x_j and the neuron h_i.

Positive-Negative Sum

For each neuron in the first layer two sums are calculated. Σ_i^+ is the sum of the input features for which the connection with the i_th_ hidden neuron has a positive weight , whereas Σ_i^- the sum of those that have a negative weight assosiated. The two sums are then compared and if the positive sum is greater than or equal to the negative the output of the neuron is 1, otherwise 0.

$$\Sigma_i^+ = \sum_{j=0}^{N-1} x_j[W1_{i,j} > 0]$$

$$\Sigma_i^- = \sum_{j=0}^{N-1} x_j[W1_{i,j} < 0]$$

$$h_i = \Sigma_i^+ \geq \Sigma_i^-$$

Sample code snippet:

```
assign positives[0] = + feature_array[1] + feature_array[2] + ...
 ↪  + feature_array[10];
assign negatives[0] = + feature_array[0] + feature_array[3] +
 ↪  feature_array[5];
assign hidden[0] = positives[0] >= negatives[0];
```

For each neuron of the output layer it's value is calculated by summing the output of hidden neurons. The binary output of the hidden neuron

$s_j$is added as-is to the sum of the output neuron y_i in the case that the weight of their connection $W2_{i,j}$ is positive and it's binary inverse is added to the sum if $W2_{i,j}$ is negative. This is equivalant to the sum of the xnor between the output vector of the hidden layer and the weight vector of the output neuron.

$$y_i = \sum_{j=0}^{N-1} \begin{cases} s_j, & \text{if } W2_{i,j} > 0 \\ \neg s_j, & \text{if } W2_{i,j} < 0 \end{cases}$$

```
assign scores[0*SUM_BITS+:SUM_BITS] = + hidden_n[0] + hidden[1] +
 ↪  hidden[2] + ... + hidden_n[39];
```