SGD-NEURAL-NET($\mathcal{D}_n, T, L, (m^1, \ldots, m^L), (f^1, \ldots, f^L)$)

for $l = 1$ **to** L
  $W^l_{ij} \sim \text{Gaussian}(0, 1/m^l)$
  $W^l_{0j} \sim \text{Gaussian}(0, 1)$
for $t = 1$ **to** T
  $i =$ random sample from $\{1, \ldots, n\}$
  $A^0 = x^{(i)}$
  // forward pass to compute the output A^L
  for $l = 1$ **to** L
    $Z^l = W^{lT} A^{l-1} + W^l_0$
    $A^l = f^l(Z^l)$
  loss $=$ Loss$(A^L, y^{(i)})$
  for $l = L$ **to** 1:
    // error back-propagation
    $\partial \text{loss}/\partial A^l =$ **if** $l < L$ **then** $\partial \text{loss}/\partial Z^{l+1} \cdot \partial Z^{l+1}/\partial A^l$ **else** $\partial \text{loss}/\partial A^L$
    $\partial \text{loss}/\partial Z^l = \partial \text{loss}/\partial A^l \cdot \partial A^l/\partial Z^l$
    // compute gradient with respect to weights
    $\partial \text{loss}/\partial W^l = \partial \text{loss}/\partial Z^l \cdot \partial Z^l/\partial W^l$
    $\partial \text{loss}/\partial W^l_0 = \partial \text{loss}/\partial Z^l \cdot \partial Z^l/\partial W^l_0$
    // stochastic gradient descent update
    $W^l = W^l - \eta(t) \cdot \partial \text{loss}/\partial W^l$
    $W^l_0 = W^l_0 - \eta(t) \cdot \partial \text{loss}/\partial W^l_0$