Отчет по лабораторной работе №1

Реализация метода обратного распространения ошибки для двуслойной полностью связанной нейронной сети

Умников Евгений

Постановка задачи

1. Изучение общей схемы метода обратного распространения ошибки.
2. Вывод математических формул для вычисления градиентов функции ошибки по параметрам нейронной сети и формул коррекции весов.
3. Проектирование и разработка программной реализации.
4. Тестирование разработанной программной реализации.

Вывод математических формул

Функции ошибки кросс энтропия:

$$E(w) = -\frac{1}{L}\sum_{k=1}^{L}\sum_{j=1}^{M} y_j^k ln(u_j^k)$$

где $y^k = (y_j^k)_{j=\overline{1,M}}$ – множество обучающих примеров,

$u^k = (u_j^k)_{j=\overline{1,M}}$ - выход нейронной сети.

SoftMax – функция активации на последнем слое:

$$\varphi(u_j) = \frac{e^{u_j}}{\sum_{j=1}^{M} e^{u_j}}.$$

Функция активации на скрытом слое – гиперболический тангенс.

$$\varphi(f_s) = th(f_s)$$

$$E(w) = -\sum_{j=1}^{M} y_j \ln\left(\frac{e^{u_j}}{\sum_{j=1}^{M} e^{u_j}}\right) = -\sum_{j=1}^{M} y_j (u_j - ln\sum_{m=1}^{M} e^{u_j}),$$

$$u_j = \sum_{s=0}^{K} w_{sj}^{(2)} \varphi\left(\sum_{i=0}^{N} w_{is}^{(1)} x_i\right).$$

$$\frac{\partial E(w)}{\partial w_{sj}^{(2)}} = \frac{\partial E}{\partial e^{u_j}} \frac{\partial e^{u_j}}{\partial w_{sj}^{(2)}}, \quad \frac{\partial e^{u_j}}{\partial w_{sj}^{(2)}} = v_s,$$

$$\delta_j^{(2)} = \frac{\partial E}{\partial e^{u_j}} = \left(\sum_{j=1}^{M} y_j\right) \cdot \frac{e^{u_j}}{\sum_{m=1}^{M} e^{u_m}} - y_j = \frac{e^{u_j}}{\sum_{m=1}^{M} e^{u_m}} - y_j$$

$$\frac{\partial E(w)}{\partial w_{is}^{(1)}} = \frac{\partial E}{\partial f_s} \frac{\partial f_s}{\partial w_{is}^{(1)}} = \delta_s^{(1)} x_i.$$

$$\frac{\partial E}{\partial f_s} = \sum_{j=1}^{M} \frac{\partial E}{\partial u_j} \frac{\partial u_j}{\partial v_s} \frac{\partial \varphi}{\partial f_s} = \frac{\partial \varphi}{\partial f_s} \sum_{j=1}^{M} \frac{\partial E}{\partial u_j} \frac{\partial u_j}{\partial v_s} = \frac{\partial \varphi}{\partial f_s} \sum_{j=1}^{M} \delta_j^{(2)} w_{sj}^2$$

Для гиперболического тангенса:

$$\frac{\partial \varphi}{\partial f_s} = (1 - \varphi)(1 + \varphi) = (1 - v_s)(1 + v_s)$$

Программная реализация

Были реализованны следующие структуры и алгоритмы:

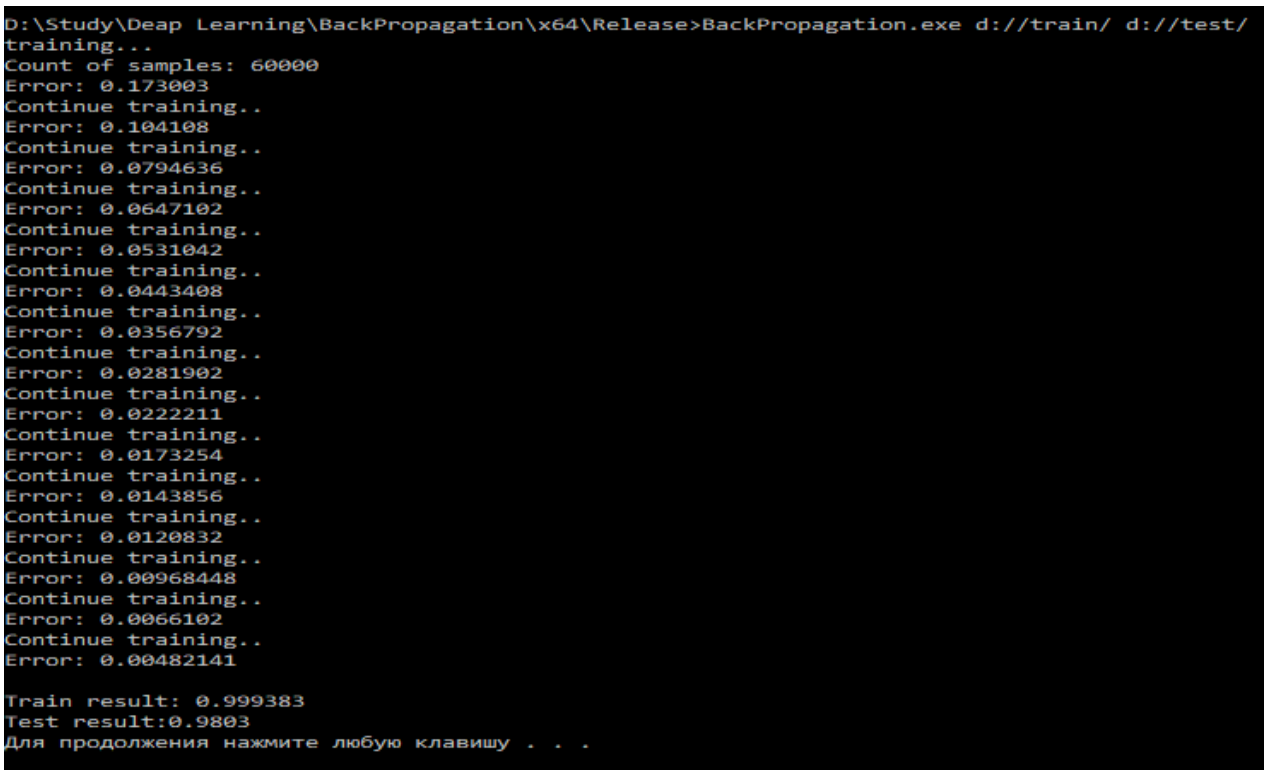
Neuron - данный класс хранит набор синапсов и значение смещения.

NeuralNetwork – класс имплементирующий двухслойную нейронную сеть. Содержит в себе два набора слоёв (std::vector<Neuron>).

BackPropagationMethod – предназначен для тренировки, а так же анализу точности нейронной сети. Для данного класса Neuron и NeuralNetwork являются friend с целью изменения параметров сети.

Результаты тестирования

Тестирование проводилось на наборе данных MNIST dataset http://yann.lecun.com/exdb/mnist. На Рис.1. показан пример работы программы.



При следующих параметрах работы

-number of epochs = 30.
-cross error = 0.005.
-learn rate = 0.01
-number of neurons in hidden layer = 300

ошибка для тренировочных данных 0.000617, для тестовых данных 0.0197.