HINT FILE: Assignment 7 (Fully Connected Neural Network)
Artificial Intelligence (CSE-241N)
IIT (BHU), Varanasi

April 14, 2018

1 Introduction

In this document we'll describe the equations for forward and backward passes (back–propogation) for a fully connected neural network. (To skip the explanation and see the algorithms themselves, go to Section 4.)

In this assignment Neural Networks will deal with supervised classification. So there are training examples consisting of input–output pairs $(\boldsymbol{x_i}_{f\times 1}, y_i)$ where $\boldsymbol{x}_i$ is the single feature vector corresponding to i^{th} training example and y_i is the class label for this feature. Further, for faster processing of data, these input output pairs are clubbed into batches, each consisting of multiple examples. A single batch looks like:

$$\boldsymbol{X} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1b} \\ x_{21} & x_{22} & \cdots & x_{2b} \\ \vdots & \vdots & \ddots & \vdots \\ x_{f1} & x_{f2} & \cdots & x_{fb} \end{pmatrix}_{f\times b} \tag{1}$$

$$\boldsymbol{Y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_b \end{pmatrix}_{b\times 1} \tag{2}$$

$\boldsymbol{X}$ can be viewed as a collection of b vectors: $(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_b)$.

Neural Networks consist of layers, each of which is composed of a weight matrix and an activation function applied elementwise, which serves to provide non–linearity to the network.

We'll first derive the gradient equations for a three layer neural network. Then we give the equations for a general sized network at the end.

2 Forward Pass

Consider a 3 layer neural network, with three weight matrices: $\boldsymbol{W}^{(i)}, i = 1, 2, 3$. Let the input be as $\boldsymbol{X}$ and output be $\boldsymbol{Y}$ of batchsize b.

The forward pass for this network would look like:

$$
\begin{aligned}
\boldsymbol{h}^{(0)} &= \boldsymbol{X}_{f\times b} && \text{(3a)}\\
\boldsymbol{a}^{(1)}_{n\times b} &= \boldsymbol{W}^{(1)}_{n\times f}\boldsymbol{h}^{(0)} && \text{(3b)}\\
\boldsymbol{h}^{(1)}_{n\times b} &= f(\boldsymbol{a}^{(1)}) && \text{(3c)}\\
\boldsymbol{a}^{(2)}_{m\times b} &= \boldsymbol{W}^{(2)}_{m\times n}\boldsymbol{h}^{(1)} && \text{(3d)}\\
\boldsymbol{h}^{(2)}_{m\times b} &= f(\boldsymbol{a}^{(2)}) && \text{(3e)}\\
\boldsymbol{a}^{(3)}_{l\times b} &= \boldsymbol{W}^{(3)}_{l\times m}\boldsymbol{h}^{(2)} && \text{(3f)}\\
\hat{\boldsymbol{Y}}_{l\times b} &= \boldsymbol{a}^{(3)} && \text{(3g)}\\
J_k &= L(Y_k, \hat{\boldsymbol{Y}}_k) && \text{(3h)}\\
J &= \sum_{i=1}^{b} J_i && \text{(3i)}
\end{aligned}
$$

where, dimension l is the number of class labels in the dataset. $\boldsymbol{h}^{(i)}$ is the output of the i^{th} hidden layer and $\boldsymbol{W}^{(i)}s$ are layer's weight matrices. f is the activation function applied on the matrices elementwise. L is the loss function which gives a single scalar output and $\hat{\boldsymbol{Y}}_k$ is the k^{th} vector of $\hat{\boldsymbol{Y}}$; k corresponds to the k^{th} example number in the batch.

Note that in the above example, the neural network has:

- Input Layer Size: f
- Hidden Layer 1 Size: n
- Hidden Layer 2 Size: m
- Output Layer Size: l

3 Backward Pass

During the packward pass, our aim is to find out the gradients of the final scalar loss J with respect to each of the weight matrices and apply gradient descent to update them. For example, we need to evaluate

$$
\nabla_{\boldsymbol{W}^{(1)}} J = \begin{pmatrix}
\frac{\partial J}{\partial W^{(1)}_{11}} & \frac{\partial J}{\partial W^{(1)}_{12}} & \cdots & \frac{\partial J}{\partial W^{(1)}_{1f}} \\
\frac{\partial J}{\partial W^{(1)}_{21}} & \frac{\partial J}{\partial W^{(1)}_{22}} & \cdots & \frac{\partial J}{\partial W^{(1)}_{2f}} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial J}{\partial W^{(1)}_{n1}} & \frac{\partial J}{\partial W^{(1)}_{n2}} & \cdots & \frac{\partial J}{\partial W^{(1)}_{nf}}
\end{pmatrix}
$$

and then update $\boldsymbol{W}^{(1)}$ using gradient descent:

$$
\boldsymbol{W}^{(1)} = \boldsymbol{W}^{(1)} - \alpha \nabla_{\boldsymbol{W}^{(1)}} J
$$

where, α is the learning rate. We have to perform this same procedure for all other matrices as well.

3.1 Backpropogation Algorithm

We'll now describe how to find out the gradient $\nabla_{\boldsymbol{W}^{(i)}} J$ for the three layer neural network considered above using backpropogation algorithm (backprop). Backprop works on the principle of chain rule of calculus. Chain rule says that if you are given $y = f(x)$ and $z = g(f(x)) = g(y)$, then $\frac{\partial z}{\partial x} = \frac{\partial z}{\partial y}\frac{\partial y}{\partial x}$, each of which can be independently calculated. Also, note that for any function g of variables x and y:

$$
\nabla g(x, y) = \frac{\partial g}{\partial x}\nabla x + \frac{\partial g}{\partial y}\nabla y \tag{4}
$$

Using these ideas we can calculate the value of $\nabla_{\boldsymbol{W}^{(3)}} J$.

3.1.1 Calculating $\nabla_{\boldsymbol{W}^{(3)}} J$

First note that

$$\begin{aligned}
\nabla_{\hat{\boldsymbol{Y}}} J &= \nabla_{\boldsymbol{a}^{(3)}} J \\
&= \begin{pmatrix} \frac{\partial J}{\partial a_{11}^{(3)}} & \frac{\partial J}{\partial a_{12}^{(3)}} & \cdots & \frac{\partial J}{\partial a_{1b}^{(3)}} \\ \frac{\partial J}{\partial a_{21}^{(3)}} & \frac{\partial J}{\partial a_{22}^{(3)}} & \cdots & \frac{\partial J}{\partial a_{2b}^{(3)}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial J}{\partial a_{l1}^{(3)}} & \frac{\partial J}{\partial a_{l2}^{(3)}} & \cdots & \frac{\partial J}{\partial a_{lb}^{(3)}} \end{pmatrix} \\
&= \begin{pmatrix} \frac{\partial J_1}{\partial a_{11}^{(3)}} & \frac{\partial J_2}{\partial a_{12}^{(3)}} & \cdots & \frac{\partial J_b}{\partial a_{1b}^{(3)}} \\ \frac{\partial J_1}{\partial a_{21}^{(3)}} & \frac{\partial J_2}{\partial a_{22}^{(3)}} & \cdots & \frac{\partial J_b}{\partial a_{2b}^{(3)}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial J_1}{\partial a_{l1}^{(3)}} & \frac{\partial J_2}{\partial a_{l2}^{(3)}} & \cdots & \frac{\partial J_b}{\partial a_{lb}^{(3)}} \end{pmatrix}
\end{aligned} \tag{5}$$

where the second equality follows from Eq. 3i. Now using the expression for $\nabla_{\boldsymbol{a}^{(3)}} J$ and Eq. 3f, we first find $\nabla_{\boldsymbol{W}^{(3)}} \boldsymbol{a}_{ij}^{(3)}$ and then calculate $\nabla_{\boldsymbol{W}^{(3)}} J$. From Eq. 3f, $a_{ij}^{(3)} = \sum_{k=1}^{m} \boldsymbol{W}_{ik}^{(3)} \boldsymbol{h}_{kj}^{(3)}$, so

$$\nabla_{\boldsymbol{W}^{(3)}} a_{ij}^{(3)} = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ h_{1j}^{(3)} & h_{2j}^{(3)} & \cdots & h_{mj}^{(3)} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix} i^{th}\text{row}, \tag{6}$$

which is a matrix with a single non–zero row. **(Can you reason out, from Eq. 3f, why $\nabla_{\boldsymbol{W}^{(3)}} a_{ij}^{(3)}$ has a single non–zero row?)** From Eq. 3h, J_k can be seen as a function of $a_{ik}^{(3)} s$. Combining this fact with Eq. 4 we get:

$$\begin{aligned}
\nabla_{\boldsymbol{W}^{(3)}} J_k &= \sum_{i=1}^{l} \left(\frac{\partial J_k}{\partial a_{ik}^{(3)}} \right) \nabla_{\boldsymbol{W}^{(3)}} a_{ik}^{(3)} \\
&= \begin{pmatrix} \frac{\partial J_k}{\partial a_{1k}^{(3)}} h_{1k}^{(3)} & \frac{\partial J_k}{\partial a_{1k}^{(3)}} h_{2k}^{(3)} & \cdots & \frac{\partial J_k}{\partial a_{1k}^{(3)}} h_{mk}^{(3)} \\ \frac{\partial J_k}{\partial a_{2k}^{(3)}} h_{1k}^{(3)} & \frac{\partial J_k}{\partial a_{2k}^{(3)}} h_{2k}^{(3)} & \cdots & \frac{\partial J_k}{\partial a_{2k}^{(3)}} h_{mk}^{(3)} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial J_k}{\partial a_{lk}^{(3)}} h_{1k}^{(3)} & \frac{\partial J_k}{\partial a_{lk}^{(3)}} h_{2k}^{(3)} & \cdots & \frac{\partial J_k}{\partial a_{lk}^{(3)}} h_{mk}^{(3)} \end{pmatrix} \\
&= \begin{pmatrix} \frac{\partial J_k}{\partial a_{1k}^{(3)}} \\ \frac{\partial J_k}{\partial a_{2k}^{(3)}} \\ \vdots \\ \frac{\partial J_k}{\partial a_{lk}^{(3)}} \end{pmatrix} \begin{pmatrix} h_{1k}^{(3)} & h_{2k}^{(3)} & \cdots & h_{mk}^{(3)} \end{pmatrix} \\
&= (\nabla_{\boldsymbol{a}^{(3)}} J)_k \boldsymbol{h}_k^{(3)^\top}
\end{aligned} \tag{7}$$

where $(\nabla_{\boldsymbol{a}^{(3)}} J)_k \boldsymbol{h}_k^{(3)^\top}$ is the outer product of $(\nabla_{\boldsymbol{a}^{(3)}} J)_k$, the k^{th} column of $\nabla_{\boldsymbol{a}^{(3)}} J$ and $\boldsymbol{h}_k^{(3)^\top}$, the k^{th} column of $\boldsymbol{h}^{(3)}$ transposed. Again, using Eq. 3i, we get:

$$\begin{aligned}\nabla_{\boldsymbol{W}^{(3)}} J &= \sum_{k=1}^{b} \nabla_{\boldsymbol{W}^{(3)}} J_k \\ &= \sum_{k=1}^{b} (\nabla_{\boldsymbol{a}^{(3)}} J)_k \boldsymbol{h}_k^{(3)^\top} \\ &= (\nabla_{\boldsymbol{a}^{(3)}} J) \boldsymbol{h}^{(3)^\top} \end{aligned} \tag{8}$$

where the last equality follows from viewing product of the two matrices as the sum of the outer products of their individual columns and rows. Eq. 8 gives the final expression for $\nabla_{\boldsymbol{W}^{(3)}} J$ and we are done.

Our next task is to calculate $\nabla_{\boldsymbol{W}^{(2)}} J$, for which we must obtain the expression for $\nabla_{\boldsymbol{a}^{(2)}} J$, and then using Eq. 3d we can proceed as we did in this section. Calculating $\nabla_{\boldsymbol{a}^{(2)}} J$ itself requires the knowledge of $\nabla_{\boldsymbol{h}^{(2)}} J$.

3.1.2 Calculating $\nabla_{\boldsymbol{h}^{(2)}} J$

Now we move on to calculate the gradient for $\boldsymbol{h}^{(2)}$. Again, from Eq. 3f: $a_{ij}^{(3)} = \sum_{k=1}^{m} \boldsymbol{W}_{ik}^{(3)} \boldsymbol{h}_{kj}^{(3)}$, so

$$\nabla_{\boldsymbol{h}^{(2)}} a_{ik}^{(3)} = \begin{array}{c} \\ \\ \\ \\ \end{array} \overset{\overset{k^{th}\text{ column}}{\downarrow}}{\begin{pmatrix} 0 & 0 & \cdots & W_{i1}^{(3)} & \cdots & 0 \\ 0 & 0 & \cdots & W_{i2}^{(3)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & W_{im}^{(3)} & \cdots & 0 \end{pmatrix}} \tag{9}$$

which is a matrix with a single non–zero column. **(Can you see why $\nabla_{\boldsymbol{h}^{(2)}} a_{ik}^{(3)}$ has a single non–zero column? Also how is it different from $\nabla_{\boldsymbol{W}^{(3)}} a_{ij}^{(3)}$ which had a single non–zero row?)**From Eq. 3h and Eq. 3f, J_k can be seen as a function of $h_{ik}^{(2)} s$. Combining this fact with Eq. 4 we get:

$$\begin{aligned}\nabla_{\boldsymbol{h}^{(2)}} J_k &= \sum_{i=1}^{l} \left(\frac{\partial J_k}{\partial a_{ik}^{(3)}} \right) \nabla_{\boldsymbol{h}^{(2)}} a_{ik}^{(3)} \\ &= \begin{pmatrix} 0 & 0 & \cdots & \sum_{i=1}^{l} W_{i1}^{(3)} \frac{\partial J_k}{\partial a_{ik}^{(3)}} & \cdots & 0 \\ 0 & 0 & \cdots & \sum_{i=1}^{l} W_{i2}^{(3)} \frac{\partial J_k}{\partial a_{ik}^{(3)}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sum_{i=1}^{l} W_{im}^{(3)} \frac{\partial J_k}{\partial a_{ik}^{(3)}} & \cdots & 0 \end{pmatrix} \end{aligned} \tag{10}$$

Again, from Eq. 3i we have $\nabla_{\boldsymbol{h}^{(2)}} J = \sum_{k=1}^{b} \nabla_{\boldsymbol{h}^{(2)}} J_k$. Using Eq. 10 with this expression we can write $(\nabla_{\boldsymbol{h}^{(2)}} J)_{ik} = \sum_{j=1}^{l} W_{ij}^{(3)} (\nabla_{\boldsymbol{a}^{(3)}} J)_{jk}$. This is equivalent to:

$$\nabla_{\boldsymbol{h}^{(2)}} J = \boldsymbol{W}^{(3)^\top} (\nabla_{\boldsymbol{a}^{(3)}} J) \tag{11}$$

3.1.3 Calculating $\nabla_{\boldsymbol{a}^{(2)}} J$

We can now calculate $\nabla_{a^{(2)}} J$. From Eq. 3e, $h_{ij}^{(2)} = f(a_{ij}^{(2)})$. So,

$$\begin{aligned}\frac{\partial J}{\partial a_{ij}^{(2)}} &= \frac{\partial J}{\partial h_{ij}^{(2)}} \frac{\partial h_{ij}^{(2)}}{\partial a_{ij}^{(2)}} \\ &= \frac{\partial J}{\partial h_{ij}^{(2)}} f'(a_{ij}^{(2)}) \end{aligned}$$

Writing the above equation in the matrix form:

$$\nabla_{\boldsymbol{a}^{(2)}} J = \nabla_{\boldsymbol{h}^{(2)}} J \odot f'(\boldsymbol{a}^{(2)}) \tag{12}$$

where $\odot$ represents elementwise multiplication.

3.1.4 Calculating $\nabla_{\boldsymbol{W}^{(2)}} J$ and $\nabla_{\boldsymbol{W}^{(1)}} J$

Now that we have the expression for $\nabla_{\boldsymbol{a}^{(2)}} J$, we can use Eq. 3d to calculate $\nabla_{\boldsymbol{W}^{(2)}} J$, using a similar procedure as adopted by us in Section 3.1.1. We can then use Eq. 3d and 3c to calculate $\nabla_{\boldsymbol{a}^{(1)}} J$, as done in Section 3.1.3. We keep on repeating this procedure till we have finally evaluated $\nabla_{\boldsymbol{W}^{(1)}} J$, at which point we are done.

4 Algorithmic form for forward pass and backward pass

In this section we extend the ideas discussed in the previous sections to an l layered fully connected neural network. The algorithms below are a modified form of those given in [1].

4.1 Forward Pass

Algorithm 1 Forward pass of a fully connected neural network

Input: Network depth, l
Input: $\boldsymbol{W}^{(i)}, i \in 1, \dots l$, the weight matrices
Input: $\boldsymbol{b}^{(i)}, i \in 1, \dots l$, the bias terms
Input: $\boldsymbol{X}_{f \times b}$, the input to the model, where f is the number of features in a single sample and b is the batchsize
Input: $\boldsymbol{Y}_{b \times 1}$, the output class labels
$\boldsymbol{h}^{(0)} = \boldsymbol{X}$
for $k = 1, \dots, l-1$ **do**
    $\boldsymbol{a}^{(k)} = \boldsymbol{W}^{(k)} \boldsymbol{h}^{(k-1)} + \boldsymbol{b}^{(k)}$
    $\boldsymbol{h}^{(k)} = f(\boldsymbol{a}^{(k)})$
$\boldsymbol{a}^{(l)} = \boldsymbol{W}^{(l)} \boldsymbol{h}^{(l-1)} + \boldsymbol{b}^{(l)}$
$\hat{\boldsymbol{Y}} = \boldsymbol{a}^{(l)}$
$J_k = L(\boldsymbol{Y}_k, \hat{\boldsymbol{Y}}_k)$, for each k in $1, \dots, b$
$J = \sum_{k=1}^{b} J_k$

4.2 Backward Pass

Algorithm 2 Backward pass of a fully connected neural network

$\boldsymbol{g} \leftarrow \nabla_{\hat{\boldsymbol{Y}}} J$
$\boldsymbol{O} = \begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix}^\top$, the number of 1s in $\boldsymbol{O}$ is equal to the number of columns in $\boldsymbol{g}$
$\nabla_{\boldsymbol{b}^{(l)}} J = \boldsymbol{g}\, \boldsymbol{O}$
$\nabla_{\boldsymbol{W}^{(l)}} J = \boldsymbol{g}\, {\boldsymbol{h}^{(l-1)}}^\top$
$\boldsymbol{g} \leftarrow \nabla_{\boldsymbol{h}^{(l-1)}} J = {\boldsymbol{W}^{(l)}}^\top \boldsymbol{g}$
for $k = l-1, \dots, 1$ **do**
    $\boldsymbol{g} \leftarrow \nabla_{\boldsymbol{a}^{(k)}} J = \boldsymbol{g} \odot f'(\boldsymbol{a}^{(k)})$
    $\boldsymbol{O} = \begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix}^\top$, the number of 1s in $\boldsymbol{O}$ is equal to the number of columns in $\boldsymbol{g}$
    $\nabla_{\boldsymbol{b}^{(k)}} J = \boldsymbol{g}\, \boldsymbol{O}$
    $\nabla_{\boldsymbol{W}^{(k)}} J = \boldsymbol{g}\, {\boldsymbol{h}^{(k-1)}}^\top$
    $\boldsymbol{g} \leftarrow \nabla_{\boldsymbol{h}^{(k-1)}} J = {\boldsymbol{W}^{(k)}}^\top \boldsymbol{g}$

5 Loss function and the activation function

In this section we would talk about the form of loss function and the activation functions we'll use in the assignment and the procedure for calculating their respective gradients.

The loss function we will work with in the assignment is the **Cross-Entropy Loss**, defined by:

$$J_k = L(Y_k, \hat{\boldsymbol{Y}}_k) = -\log\left(\frac{e^{\hat{Y}_{Y_k,k}}}{\sum_{i=1}^{l} e^{\hat{Y}_{i,k}}}\right) \tag{13}$$

where J_k is the loss for the k^{th} sample, l is the number of classes and Y_k is the correct label (a single scalar number) corresponding to the class of the k^{th} sample. Derivative of the loss function with respect to $\hat{Y}_{ik}$ is given by:

$$\frac{\partial J_k}{\partial \hat{Y}_{ik}} = \begin{cases} \frac{e^{\hat{Y}_{ik}}}{\sum_{i=1}^{l} e^{\hat{Y}_{ik}}} - 1 & , i = Y_k \\ \frac{e^{\hat{Y}_{ik}}}{\sum_{i=1}^{l} e^{\hat{Y}_{ik}}} & , i \neq Y_k \end{cases} \tag{14}$$

For further information, read the section "Softmax classifier" and "Practical issues: Numeric stability" on `http://cs231n.github.io/linear-classify/#softmax`.

One of the popular activation functions is tanh(). Again, although you are free to use any activation function of your choice in the assignment, we'll describe just tanh() here.

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{15}$$

Its derivative is

$$\frac{\partial \tanh(x)}{\partial x} = 1 - \tanh^2(x) \tag{16}$$

Remember to apply the activation functions elementwise, and also use `numpy.tanh()` function instead of writing your own implementations.

References

[1] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.