Backpropagation Computations in Matrix Form

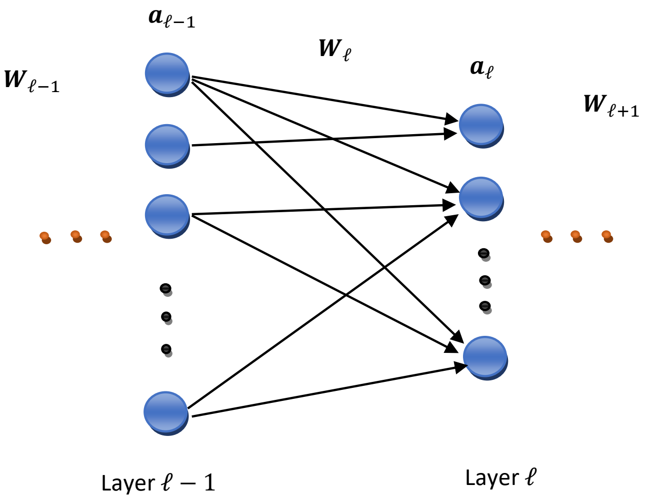
Mostafa Sadeghi

Electrical Engineering Department
Sharif University of Technology
Tehran, Iran.

February 2018

Contents

L-layer neural network



L-layer neural network

An L-layer artificial neural network with input-output $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N}$ can be described by the following equations ($\ell = 1, \ldots, L$):

$$\begin{cases} \mathbf{z}_\ell = \mathbf{W}_\ell \mathbf{a}_{\ell-1} + \mathbf{b}_\ell \\ \mathbf{a}_\ell = \sigma_\ell(\mathbf{z}_\ell) \end{cases}$$

- $\mathbf{a}_\ell$ = input to the ℓth layer
- $\mathbf{a}_0 = \mathbf{x}$ (input) and $\mathbf{a}_L$ = estimation of $\mathbf{y}$ (output)
- $\mathbf{W}_\ell$ = weighting matrix connecting layer $\ell - 1$ to layer ℓ
- $\mathbf{b}_\ell$ = bias vector for layer ℓ
- $\sigma_\ell(.)$ = component-wise nonlinear activation function of layer ℓ

Overall training cost function

The overall training cost function can be written as:

$$C_L = \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{a}_L^i - \mathbf{y}_i\|_2^2$$

where, $\mathbf{a}_L^i$ is the network output due to input $\mathbf{x}_i$. For simplicity, we only consider the following single cost, for a generic input-output $(\mathbf{x}, \mathbf{y})$, to derive the gradients with respect to network's parameters. The obtained expressions can easily be extended to mini-batches of data.

$$c_L = \|\mathbf{a}_L - \mathbf{y}\|_2^2$$

*Note. Henceforth, we do not make any difference between *derivative* and *gradient*.

Sanity checks for derivative

Dimension check

- The derivative of $y = f(\mathbf{x})$ with respect to $\mathbf{x}$ is of the same dimension as $\mathbf{x}$.
- To see if a derivative is correct, always check dimension compatibility for matrix or vector multiplications.

  Example. $\mathbf{A} \in \mathbb{R}^{m\times n}$ and $\mathbf{x} \in \mathbb{R}^n$:

$$\nabla_{\mathbf{x}} \|\mathbf{A}\mathbf{x}\|_2^2 = \begin{cases} 2\mathbf{A}\mathbf{x} & \text{Different dimension than } \mathbf{x} \\ 2\mathbf{A}\mathbf{A}^T\mathbf{x} & \text{Incompatible dimension} \\ 2\mathbf{A}^T\mathbf{A}\mathbf{x} & \text{Correct} \end{cases}$$

Numerical check

- For complex derivative expressions, a useful sanity check is to compare it with numerical derivative:

$$\frac{\partial f}{\partial x_i} \simeq \frac{f(x_i + h) - f(x_i - h)}{2h}, \quad \forall i$$

  for a small $h > 0$. Here, x_i denotes the ith entry of $\mathbf{x}$.

A useful trick for gradient computation

Consider a general expression like this:

$$f(\mathbf{W}, \mathbf{X}, \mathbf{V}) = \|\mathbf{W}\mathbf{X}\mathbf{V} - \mathbf{U}\|_F^2$$

To compute the gradient with respect to each variable, note that the transposed of the other variables surrounding it are multiplied from the same side as they are. That is:

$$\begin{cases} \nabla_{\mathbf{W}} f = -2(\mathbf{W}\mathbf{X}\mathbf{V} - \mathbf{U})(\mathbf{X}\mathbf{V})^T \\ \nabla_{\mathbf{X}} f = -2\mathbf{W}^T(\mathbf{W}\mathbf{X}\mathbf{V} - \mathbf{U})\mathbf{V}^T \\ \nabla_{\mathbf{V}} f = -2(\mathbf{W}\mathbf{X})^T(\mathbf{W}\mathbf{X}\mathbf{V} - \mathbf{U}) \end{cases}$$

Example: $f(\mathbf{x}, \mathbf{W}) = \|\mathbf{y} - \mathbf{W}\mathbf{x}\|_2^2$

$$\begin{cases} \nabla_{\mathbf{x}} f = -2\mathbf{W}^T(\mathbf{y} - \mathbf{W}\mathbf{x}) \\ \nabla_{\mathbf{W}} f = -2(\mathbf{y} - \mathbf{W}\mathbf{x})\mathbf{x}^T \end{cases}$$

Derivatives for output layer

Backprop begins from the output layer and computes derivatives in a backward manner (a forward pass is firstly performed to update parameter values):

$$\begin{cases} \frac{\partial c_L}{\partial \mathbf{a}_L} = 2(\mathbf{a}_L - \mathbf{y}) \\ \frac{\partial c_L}{\partial \mathbf{z}_L} = \frac{\partial c_L}{\partial \mathbf{a}_L} \cdot \frac{\partial \mathbf{a}_L}{\partial \mathbf{z}_L} = 2(\mathbf{a}_L - \mathbf{y}) \odot \sigma'_L(\mathbf{z}_L) \\ \frac{\partial c_L}{\partial \mathbf{W}_L} = \frac{\partial c_L}{\partial \mathbf{z}_L} \cdot \frac{\partial \mathbf{z}_L}{\partial \mathbf{W}_L} = \frac{\partial c_L}{\partial \mathbf{z}_L} \cdot \mathbf{a}_{L-1}^T \\ \frac{\partial c_L}{\partial \mathbf{b}_L} = \frac{\partial c_L}{\partial \mathbf{z}_L} \cdot \frac{\partial \mathbf{z}_L}{\partial \mathbf{b}_L} = \frac{\partial c_L}{\partial \mathbf{z}_L} \end{cases}$$

To compute the derivatives of the previous layers recursively, let's define

$\boxed{\boldsymbol{\delta}_\ell \triangleq \frac{\partial c_\ell}{\partial \mathbf{z}_\ell}}$ This is called the sensitivity vector of layer ℓ.

Derivatives for intermediate layers

Recall:

$$\begin{cases} \mathbf{z}_\ell = \mathbf{W}_\ell \mathbf{a}_{\ell-1} + \mathbf{b}_\ell \\ \mathbf{a}_\ell = \sigma_\ell(\mathbf{z}_\ell) \end{cases}$$

Then, for $\ell = L-1, \dots, 1$:

$$\begin{cases} \frac{\partial c_L}{\partial \mathbf{a}_\ell} = \frac{\partial c_L}{\partial \mathbf{z}_{\ell+1}} \cdot \frac{\partial \mathbf{z}_{\ell+1}}{\partial \mathbf{a}_\ell} = \mathbf{W}_{\ell+1}^T \boldsymbol{\delta}_{\ell+1} \\ \frac{\partial c_L}{\partial \mathbf{z}_\ell} = \frac{\partial c_L}{\partial \mathbf{a}_\ell} \cdot \frac{\partial \mathbf{a}_\ell}{\partial \mathbf{z}_\ell} = (\mathbf{W}_{\ell+1}^T \boldsymbol{\delta}_{\ell+1}) \odot \sigma_\ell'(\mathbf{z}_\ell) = \boldsymbol{\delta}_\ell \\ \frac{\partial c_L}{\partial \mathbf{W}_\ell} = \frac{\partial c_L}{\partial \mathbf{z}_\ell} \cdot \frac{\partial \mathbf{z}_\ell}{\partial \mathbf{W}_\ell} = \boldsymbol{\delta}_\ell \mathbf{a}_{\ell-1}^T \\ \frac{\partial c_L}{\partial \mathbf{b}_\ell} = \frac{\partial c_L}{\partial \mathbf{z}_\ell} \cdot \frac{\partial \mathbf{z}_\ell}{\partial \mathbf{b}_\ell} = \boldsymbol{\delta}_\ell \end{cases}$$

Final expressions

- (Forward pass) For $\ell = 1, \dots, L$, compute

$$\begin{cases} \mathbf{z}_\ell = \mathbf{W}_\ell \mathbf{a}_{\ell-1} + \mathbf{b}_\ell \\ \mathbf{a}_\ell = \sigma_\ell(\mathbf{z}_\ell) \end{cases}$$

- (Backward pass) Set $\boldsymbol{\delta}_L = 2(\mathbf{a}_L - \mathbf{y}) \odot \sigma'_L(\mathbf{z}_L)$. For $\ell = L-1, \dots, 1$ compute:

  - $\boldsymbol{\delta}_\ell = (\mathbf{W}_{\ell+1}^T \boldsymbol{\delta}_{\ell+1}) \odot \sigma'_\ell(\mathbf{z}_\ell)$
  - $\frac{\partial c_L}{\partial \mathbf{W}_\ell} = \boldsymbol{\delta}_\ell \mathbf{a}_{\ell-1}^T$
  - $\frac{\partial c_L}{\partial \mathbf{b}_\ell} = \boldsymbol{\delta}_\ell$

  Update parameters using gradient descent:

$$\begin{cases} \mathbf{W}_\ell \leftarrow \mathbf{W}_\ell - \alpha \frac{\partial c_L}{\partial \mathbf{W}_\ell} \\ \mathbf{b}_\ell \leftarrow \mathbf{b}_\ell - \alpha \frac{\partial c_L}{\partial \mathbf{b}_\ell} \end{cases}$$