Backpropagation: Practice Questions

Paweł A. Pierzchlewicz

August 11, 2020

This is an addition to the article on medium. I would highly recommend to try solving these questions to accompany you in understanding the backpropagation algorithm. You don't have to solve all of them, however each question bases upon the knowledge gained from the previous ones. The implementation questions provide you with sample models, but feel free to play around with them to better understand the models capabilities. If you get stuck the article should help you get through the problems.

This is a step by step guide to derive and understand the backpropagation algorithm. These are the steps I personally followed to understand the reasoning behind the algorithm. My idea was to start from the simplest model possible and slowly build up to more complex cases, while maintaining a full understanding of every step.

1 Mathematical Primer

1.1 Linear Algebra

1.1.1

Compute the value of y

a)

$$y = \begin{bmatrix} w_0 & w_1 \end{bmatrix} \begin{bmatrix} 1 \\ x \end{bmatrix}$$

b)

$$y = \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} \begin{bmatrix} 1 & x \end{bmatrix}$$

c)

$$y = \begin{bmatrix} 1 & x \end{bmatrix} \begin{bmatrix} w_{0,0} & w_{0,1} \\ w_{1,0} & w_{1,1} \end{bmatrix}$$

1.1.2

Rewrite the following equations in vector form

a)

$$y = \sum_{n}^{N} w_n x_n$$

b)

$$y = \sum_{n}^{N} w_n x_n - y_n$$

c)

$$y = \sum_{n}^{N} (w_n x_n - y_n) x_n$$

1.2 Calculus

1.2.1

Find the derivative of $L(x)$ with respect to x

$$L(x) = (x - a)^2$$

1.2.2

Find the derivative of $g(x)$ with respect to x

$$g(x) = \frac{1}{1 + x^2}$$

1.2.3

What is the derivative of

a)

$$h(x) = (f \circ g)(x)$$

b)

$$z(x) = (f \circ g \circ h)(x)$$

2 Linear Regression

2.1 Model without bias

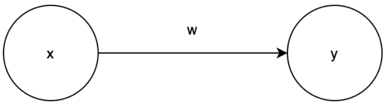

Figure 1: **Linear Model Without Bias**

2.1.1

You want to fit the linear model:

$$\hat{y}(x) = wx$$

to a dataset of labels Y and values X. To do so we minimise the mean squared error

$$y(x) = \frac{1}{2N}\sum_{n}^{N}(\hat{y}(x_n) - y_n)^2$$

a)
Compute the derivative $\frac{\partial L(X;w)}{\partial w}$

b)
Rewrite the derivative in vector form.

c)
Write the gradient descent update rule.

2.1.2

Simulate a dataset consisting of N samples using the following equation:

$$y(x) = -x + \mathcal{N}(0,1)$$

where $\mathcal{N}(0,1)$ is a sample from the normal distribution. Using NumPy you can sample from this normal distribution using *numpy.random.normal(loc=0, scale=1, size=N)* which will generate N samples. Then fit the linear model to the dataset using the gradient descent update rule.

2.2 Model with bias

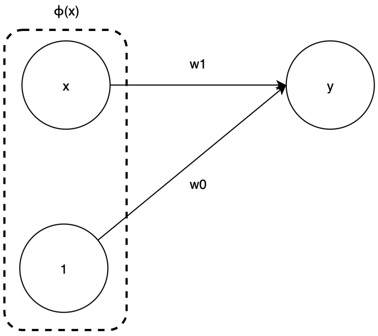

Figure 2: **Linear Model with Bias**

2.2.1

Find the feature vector $\phi(x)$ that satisfies:

$$w_1x + w_2 = \mathbf{w}^\top \phi(x)$$

2.2.2

Write the gradient descent update rule for the following model:

$$y(x) = \mathbf{w}^\top \phi(x)$$

using the steps from question 2.1.1.

2.2.3

Implement the model to fit a simulated dataset which was generated using:

$$y(x) = 3x + 5 + \mathcal{N}(0, 1)$$

3 Logistic Regression

3.1 Activation Functions

3.1.1

For logistic regression, we define an activation function $g(x)$, which is the logistic function.

$$g(x) = \frac{1}{1 + e^{-x}}$$

Calculate the derivative of $g(x)$.

3.1.2 (optional)

There is a whole family of activation functions. Calculate the derivatives of a couple:

a) ReLU

$$g(x) = [x]_+ = \max(0, x)$$

b) Heaviside

$$g(x) = \mathbf{1}[x] = \begin{cases} 1 & x > 0 \\ 0 & \text{otherwise} \end{cases}$$

c) Linear

$$g(x) = x$$

3.2 Perceptron

3.2.1

Activation functions operate on the output of the layer, therefore the activation becomes $a(x) = g(\hat{y})$. Given this we can find that the new loss function is

$$L(w) = \frac{1}{2N} \sum_{n}^{N} (g(\hat{y}(x_n)) - y_n)^2$$

compute it's derivative and write it in vector form and the resultant update rule.

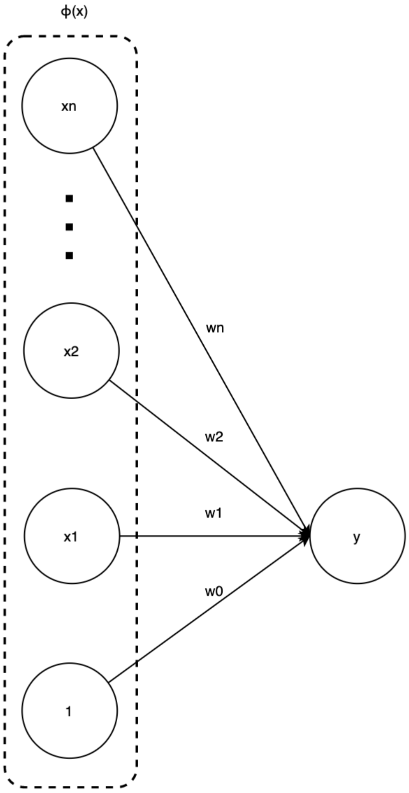

Figure 3: **Multiple inputs and one output model**

3.2.2

Simulate two classes using a multivariate normal distribution. In python this can be accomplished using *numpy.random.multivariate_normal* method. As parameters for the distributions use:

$$\mu_1 = \begin{bmatrix} 5 \\ 5 \end{bmatrix}$$

$$\mu_2 = \begin{bmatrix} 7 \\ 7 \end{bmatrix}$$

$$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Implement a perceptron model with a sigmoid (logistic) activation function, using the derived update rule.