Artificial Neural Networks Via Back Propagation For the Iris Data

Sunny Lee

July 31, 2022

Introduction

- What is an Artificial Neural Network (ANN)?

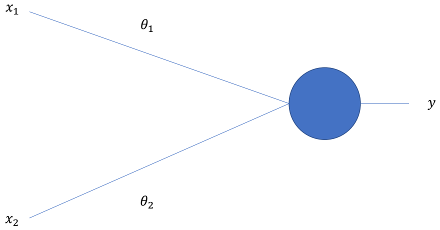
Introduction

- What is an Artificial Neural Network (ANN)?
  - Classification

Introduction

- What is an Artificial Neural Network (ANN)?
  - Classification
  - Regression

Perceptron



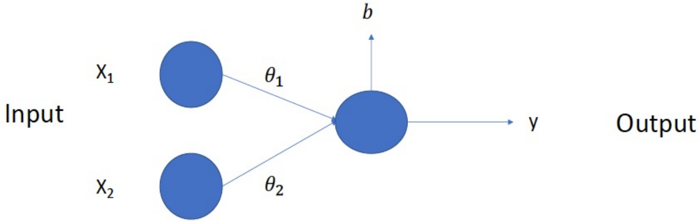
Sigmoid Activation

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

Sigmoid Activation

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

Basic Network



Forward Pass

Dot Product:

$$z_k^{(i)} = \theta_0 x_0^{(i)} + \theta_1 x_1^{(i)} + \theta_2 x_2^{(i)} + \cdots + \theta_j x_j^{(i)}$$

Applying the activation function:

$$a_k^{(i)} = g(z_k^{(i)})$$

Loss Function

- Mean Squared Error:

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(a^{(i)} - y^{(i)})^2$$

Gradient Descent

Gradient descent iteration:

$$\theta_n = \theta_{n-1} - \alpha \frac{\partial J(\theta)}{\partial \theta_{n-1}}$$

Loss Function Derivative

- Derivative:
  - $$\frac{\partial}{\partial \theta_j}\left(\frac{1}{2m}\sum_{i=1}^{m}(a^{(i)} - y^{(i)})^2\right)$$

Loss Function Derivative

- Derivative:
  - $$\frac{\partial}{\partial\theta_j}\left(\frac{1}{2m}\sum_{i=1}^{m}(a^{(i)}-y^{(i)})^2\right)$$
  - $$\frac{1}{2m}\sum_{i=1}^{m}\frac{\partial}{\partial\theta_j}(a^{(i)}-y^{(i)})^2$$

Loss Function Derivative

- Derivative:
  - $$\frac{\partial}{\partial \theta_j}\left(\frac{1}{2m}\sum_{i=1}^{m}(a^{(i)}-y^{(i)})^2\right)$$
  - $$\frac{1}{2m}\sum_{i=1}^{m}\frac{\partial}{\partial \theta_j}(a^{(i)}-y^{(i)})^2$$
  - $$\frac{1}{m}\sum_{i=1}^{m}(a^{(i)}-y^{(i)})\frac{\partial}{\partial \theta_j}(a^{(i)})$$

Loss Function Derivative

- Derivative:
  - $$\frac{\partial}{\partial \theta_j}\left(\frac{1}{2m}\sum_{i=1}^{m}(a^{(i)}-y^{(i)})^2\right)$$
  - $$\frac{1}{2m}\sum_{i=1}^{m}\frac{\partial}{\partial \theta_j}(a^{(i)}-y^{(i)})^2$$
  - $$\frac{1}{m}\sum_{i=1}^{m}(a^{(i)}-y^{(i)})\frac{\partial}{\partial \theta_j}(a^{(i)})$$
  - $$\frac{1}{m}\sum_{i=1}^{m}(a^{(i)}-y^{(i)})g'(z)\frac{\partial}{\partial \theta_j}z$$

Loss Function Derivative

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m}\sum_{i=1}^{m}(a^{(i)} - y^{(i)})(1 - g(z))(g(z))x_j$$

Gradient Descent Updates

$$\begin{bmatrix}\theta_0\\ \theta_1\\ \theta_2\end{bmatrix} = \begin{bmatrix}\theta_0\\ \theta_1\\ \theta_2\end{bmatrix} - \alpha \begin{bmatrix}\frac{1}{m}\sum_{i=1}^{m}(a^{(i)} - y^{(i)})(1 - g(z))(g(z))x_0\\ \frac{1}{m}\sum_{i=1}^{m}(a^{(i)} - y^{(i)})(1 - g(z))(g(z))x_1\\ \frac{1}{m}\sum_{i=1}^{m}(a^{(i)} - y^{(i)})(1 - g(z))(g(z))x_2\end{bmatrix}$$

Stochastic Gradient Descent

- Stochastic Gradient Descent
  - Small subsets of data

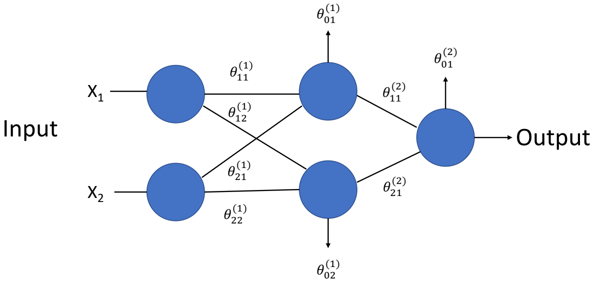
Stochastic Gradient Descent

- Stochastic Gradient Descent
  - Small subsets of data
  - Estimation of gradient

Stochastic Gradient Descent

- Stochastic Gradient Descent
  - Small subsets of data
  - Estimation of gradient
  - Quicker compute times

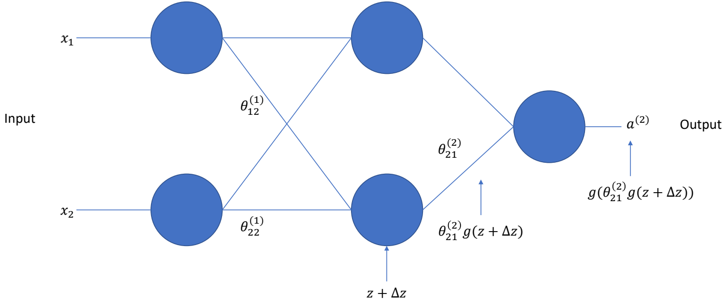
More Complicated Example



Forward Pass

$$z^l = (\theta^l)^T a^{l-1}$$
$$a^l = g(z^l)$$
$$x = a^1$$

Back Propagation



Back Propagation

$$\delta_j^l \equiv \frac{\partial J}{\partial z_j^l}$$

First Equation

- $$\frac{\partial J}{\partial z_j^L}$$

First Equation

- $$\frac{\partial J}{\partial z_j^L}$$

- $$\frac{\partial J}{\partial a_j^L} \frac{\partial a_j^L}{\partial z_j^L}$$

First Equation

- $$\frac{\partial J}{\partial z_j^L}$$
- $$\frac{\partial J}{\partial a_j^L}\frac{\partial a_j^L}{\partial z_j^L}$$
- $$\frac{1}{m}\sum_{j=1}^{m}(a_j^L - y)g'(z_j^L)$$

First Equation

- $$\frac{\partial J}{\partial z_j^L}$$
- $$\frac{\partial J}{\partial a_j^L} \frac{\partial a_j^L}{\partial z_j^L}$$
- $$\frac{1}{m} \sum_{j=1}^{m} (a_j^L - y) g'(z_j^L)$$
- $$\nabla_a J \odot g'(z)$$

Second Equation

- $$\frac{\partial J}{\partial z_j^l}$$

Second Equation

- $$\frac{\partial J}{\partial z_j^l}$$

- $$\sum_k \frac{\partial J}{\partial z_k^{l+1}} \frac{\partial z_k^{l+1}}{\partial z_j^l}$$

Second Equation

- $$\frac{\partial J}{\partial z_j^l}$$
- $$\sum_k \frac{\partial J}{\partial z_k^{l+1}} \frac{\partial z_k^{l+1}}{\partial z_j^l}$$
- $$\sum_k \delta_k^{l+1} \frac{\partial z_k^{l+1}}{\partial z_j^l}$$

Second Equation

- Since:

$$z_k^{l+1} = \theta_{jk}^l \cdot g(z_j^l) + \theta_{0k}^l$$

Second Equation

- Since:

$$z_k^{l+1} = \theta_{jk}^l \cdot g(z_j^l) + \theta_{0k}^l$$

-

$$\sum_k \delta_k^{l+1} \theta_{jk}^l g'(z_j^l)$$

Second Equation

- Since:

$$z_k^{l+1} = \theta_{jk}^l \cdot g(z_j^l) + \theta_{0k}^l$$

-

$$\sum_k \delta_k^{l+1} \theta_{jk}^l g'(z_j^l)$$

-

$$(\theta^l)^T \delta^{l+1} \odot g'(z^l)$$

Third and Fourth Equations

- $$\frac{\partial J}{\partial \theta^l_{jk}}$$

Third and Fourth Equations

- $$\frac{\partial J}{\partial \theta^{l}_{jk}}$$
- $$\frac{\partial J}{\partial z^{l}_{k}} \frac{\partial z^{l}_{k}}{\partial \theta^{l}_{jk}}$$

Third and Fourth Equations

- $$\frac{\partial J}{\partial \theta^l_{jk}}$$

- $$\frac{\partial J}{\partial z^l_k}\frac{\partial z^l_k}{\partial \theta^l_{jk}}$$

- Since:

$$z^l_k = \theta^l_{jk} \cdot g(z^{l-1}_j) + \theta^l_{0k}$$

Third and Fourth Equations

- $$\frac{\partial J}{\partial \theta^l_{jk}}$$
- $$\frac{\partial J}{\partial z^l_k}\frac{\partial z^l_k}{\partial \theta^l_{jk}}$$
- Since:
$$z^l_k = \theta^l_{jk} \cdot g(z^{l-1}_j) + \theta^l_{0k}$$
- $$\delta^l_k g(z^{l-1}_j) = \delta^l_k a^{l-1}_j$$

Third and Fourth Equations

- $$\frac{\partial J}{\partial \theta^{l}_{0k}}$$

Third and Fourth Equations

- $$\frac{\partial J}{\partial \theta_{0k}^{l}}$$

- $$\frac{\partial J}{\partial z_k^l}\frac{\partial z_k^l}{\partial \theta_{0k}^{l}}$$

Third and Fourth Equations

- $$\frac{\partial J}{\partial \theta_{0k}^{l}}$$

- $$\frac{\partial J}{\partial z_k^l} \frac{\partial z_k^l}{\partial \theta_{0k}^{l}}$$

- Since:

$$z_k^l = \theta_{jk}^{l} \cdot g(z_j^{l-1}) + \theta_{0k}^{l}$$

Third and Fourth Equations

- $$\frac{\partial J}{\partial \theta_{0k}^{l}}$$

- $$\frac{\partial J}{\partial z_k^l}\frac{\partial z_k^l}{\partial \theta_{0k}^{l}}$$

- Since:

$$z_k^l = \theta_{jk}^l \cdot g(z_j^{l-1}) + \theta_{0k}^l$$

- $$\delta_k^l$$

Back Propagation Equations

$$\delta_j^l \equiv \frac{\partial J}{\partial z_j^l}$$

$$\delta^L = \nabla_a J \odot g'(z^L)$$

$$\delta^l = (\theta^l)^T \delta^{l+1} \odot g'(z^l)$$

$$\frac{\partial J}{\partial \theta_{jk}^l} = \delta_k^l a_j^{l-1}$$

$$\frac{\partial J}{\partial \theta_{0k}^l} = \delta_k^l$$

Mini Batch Code

```
createBatch <- function(train_x, train_y){
  rows <- sample(nrow(train_x))
  train_x <- as.data.frame(train_x[rows, ])
  train_y <- as.data.frame(train_y[rows])

  mini_batches <- split(train_x, (seq(nrow(train_x))-1) %/% batch_size)
  mini_batches_y <- split(train_y, (seq(nrow(train_x))-1) %/% batch_size)

  return(list(mini_batches, mini_batches_y))
}
```

Feed Forward Code

```
feedForward <- function(w, a, z, batch){
  a1 <- as.matrix(batch)
  a[[1]] <- a1
  for (i in 2:length(layers))
  {
    z[[i]] <- as.matrix(cbind(rep(1, dim(a[[i-1]])[1]), a[[i-1]])) %*% t(as.matrix(w[[i-1]]))
    a[[i]] <- sig(z[[i]])
  }

  return(list(a, z))
}
```

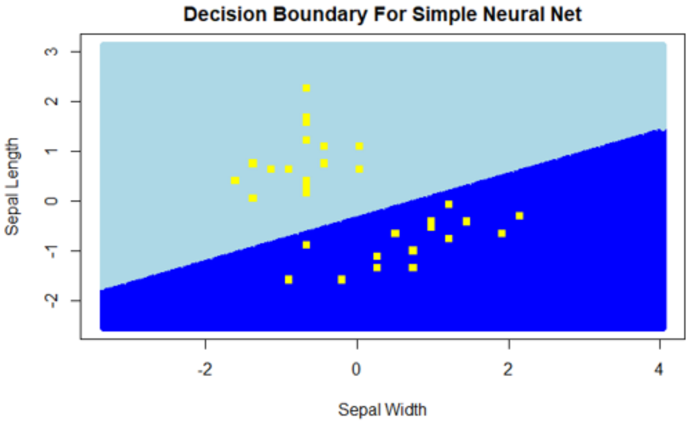
Back Propagation Code

```
backProp <- function(w, delta, a, batch_y, num_layers, batch_size, learn_rate, epsilon){
  delta[[num_layers]] <- (1/batch_size) * (a[[num_layers]]-as.matrix(batch_y)) * diffsig(z[[num_layers]])
  for (i in (num_layers-1):2)
  {
    delta[[i]] <- (1/batch_size) * (as.matrix(delta[[i+1]]) %*% (w[[i]][, -1])) * as.matrix(diffsig(z[[i]]))
  }

  for (i in length(w):1)
  {
    w_grad[[i]] <- (1/batch_size) * t(delta[[i+1]]) %*% cbind(rep(1, dim(a[[i]])[1]), a[[i]])
    if(sqrt(sum(w_grad[[i]]^2)) < epsilon){
      print("Convergence of gradients")
      return(list(-1, w))
    }
  }

  for (i in length(w):1)
  {
    w[[i]] <- w[[i]] - (learn_rate * w_grad[[i]])
  }

  return(list(1, w))
}
```

Iris Data


Setosa and Virginica Irises for Training
Sepal Length
Sepal Width
2
1
0
-1
-2
-1
0
1
2
3
Setosa
Virginica

Simple ANN Results



Simple ANN Results

	0	1
0	32	0
1	0	32

More Complicated ANN Results

	0	1
0	32	0
1	0	32

Simple ANN Nonlinear Results

test_y

	0	1
0	24	2
1	8	24

Complex ANN Nonlinear Results

test_y

	0	1
0	28	0
1	4	26

- Normalized data: [Sepal Width, Sepal Length] = $\begin{bmatrix} 0.5067965 & -0.6560779 \\ -0.6683838 & 0.1603746 \end{bmatrix}$
- Classification: Setosa = 0, Virginica 1

$$\begin{bmatrix} 1 & 0.5067965 & -0.6560779 \\ 1 & -0.6683838 & 0.1603746 \end{bmatrix} \begin{bmatrix} -0.2148802 & 2.7147823 \\ 1.3172717 & 0.7104345 \\ -2.5719430 & 0.4118125 \end{bmatrix} = \begin{bmatrix} 2.140104 & 2.804647 \\ -1.507798 & 2.305984 \end{bmatrix}$$

$$g\left(\begin{bmatrix} 2.140104 & 2.804647 \\ -1.507798 & 2.305984 \end{bmatrix}\right) = \begin{bmatrix} 0.8947404 & 0.9429264 \\ 0.1812654 & 0.9093714 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0.8947404 & 0.9429264 \\ 1 & 0.1812654 & 0.9093714 \end{bmatrix} \begin{bmatrix} 2.072452 \\ -6.001327 \\ 1.215538 \end{bmatrix} = \begin{bmatrix} -2.151015 \\ 2.089994 \end{bmatrix}$$

$$g\left(\begin{bmatrix} -2.151015 \\ 2.089994 \end{bmatrix}\right) = \begin{bmatrix} 0.1042364 \\ 0.8899269 \end{bmatrix}$$

References

- Michael A. Nielsen, "Neural Networks and Deep Learning", Determination Press, 2015
- Rosenblatt, Frank. "The perceptron: a probabilistic model for information storage and organization in the brain." Psychological review 65.6 (1958): 386.
- R. A. Fisher (1936). "The use of multiple measurements in taxonomic problems". Annals of Eugenics. 7(2): 179-188.