CIS 520, Machine Learning, Fall 2019
Homework 4

Matthew Scharf

October 8, 2019

1 Neural Networks: Backpropagation

Your task is to now compute the derivatives of the loss function given in 1 with respect to W_1, W_2, b_1 and b_2 by hand, i.e., $\frac{\partial Loss}{\partial W_1}$, $\frac{\partial Loss}{\partial b_1}$, and $\frac{\partial Loss}{\partial b_2}$.

$$Loss = y * ln(\sigma(z_2)) + (1 - y) * ln(1 - \sigma(z_2)) \tag{1}$$

Show all the intermediate derivative computation steps. You might benefit from making a rough schematic of the backpropagation process. Also recall the derivatives of the softmax function and the tanh function:

$$\frac{d\sigma(z_2)}{dz} = ln(\sigma(z_2)) * ln(1 - \sigma(z_2)) \tag{2}$$

$$\frac{\partial tanh(z_1)}{\partial z_1} = 1 - tanh^2(z_1) \tag{3}$$

For $u \in \{W_1, W_2, b_1, b_2\}$:

$\frac{\partial Loss}{\partial u} = y(1 - \sigma(z_2)\frac{\partial z_2}{\partial u} + (1 - y)(-\sigma(z_2))\frac{\partial z_2}{\partial u} = \frac{\partial z_2}{\partial u}(y - \sigma(z_2))$

Using this equation:

$$\frac{\partial Loss}{\partial W_1} = \frac{\partial z_2}{\partial W_1}(y - \sigma(z_2)) = W_2(1 - tanh^2(z_1))x(y - \sigma(z_2)) \tag{4}$$

$$\frac{\partial Loss}{\partial W_2} = \frac{\partial z_2}{\partial W_2}(y - \sigma(z_2)) = a_1(y - \sigma(z_2)) \tag{5}$$

$$\frac{\partial Loss}{\partial b_1} = \frac{\partial z_2}{\partial b_1}(y - \sigma(z_2)) = (y - \sigma(z_2)) \tag{6}$$

$$\frac{\partial Loss}{\partial b_2} = \frac{\partial z_2}{\partial b_2}(y - \sigma(z_2)) = W_2(1 - tanh^2(z_1))(y - \sigma(z_2)) \tag{7}$$

2 Programming

2.1 Random Forests

Tabulate the prediction results on the test set in Table 1.

n_estimators	**Accuracy (%)**
1	76.4
10	94.6
50	96.6
100	97.0
500	97.3

Table 1: Accuracy for the Random Forests classification problem on the test set

2.2 Kernel SVM

Tabulate the prediction results on the test set in Table 2.

kernel	**Accuracy (%)**
Linear	98.3
Poly	98.7
RBF	44.8

Table 2: Accuracy for the kernel SVM classification problem on the test set

2.3 Multi Layer Perceptron

Tabulate the prediction results on the test set in Table 3.

Network Architecture	**Accuracy (%)**
(3)	72.1
(10)	94.3
(10,10,10)	94.6
(20,50,20)	96.3

Table 3: Accuracy for the MLP classification problem on the test set

2.4 Short Answer

For the random forest and multi layer perceptron, we see that the the accuracy improves from 70's to 90's as we increase the complexity. This is consistent with what we would expect.

For the Kernel SVM, the linear and polynomial kernels seem to work well. The radial basis function, however, seems to work considerably less well. This indicates that the correct decision boundary is best described by a polynomial function, and the radial basis function is an unnecessarily complicated transformation of the space.

3 Convolutional Neural Networks

3.1 Theory

1. The output volume is: 29x29x32

...

2. Including bias parameters, this hidden layer has the following number of parameters: ((300*300*3)*100)+1=27,000,001

...

3. The output from the simple convolutional layer for the given filter and input is:

Row	Column	Filter	Value
1	1	1	9
1	1	2	20
1	2	1	15
2	1	1	18

Table 4: Output from convolution

4. The total number of trainable parameters in this network is:

$$(30 * 20 + 1) + (8 * (20 * 20 + 1)) + (20 + 1) = 601 + 3208 + 21 = 3830$$

5. State two advantages of convolutional neural networks over fully connected networks.

   They take advantage of translational invariance (arrangements of pixels should represent the same thing regardless of where they are in the picture) as well as the importance of locality in pictures (the closer pixels are to each other the more they should combined to find meaning). Fully connected networks do not have this property and so are more complex/harder to train.

3.2 Programming

For this question, refer to the Jupyter Notebook. You will be using PyTorch to implement a convolutional neural network – the notebook will have detailed instructions. We will be using the fashion MNIST dataset for a classification task.

3.2.1 Convolutional Neural Network

Add the accuracy and the loss curve from tensorboard in this report:

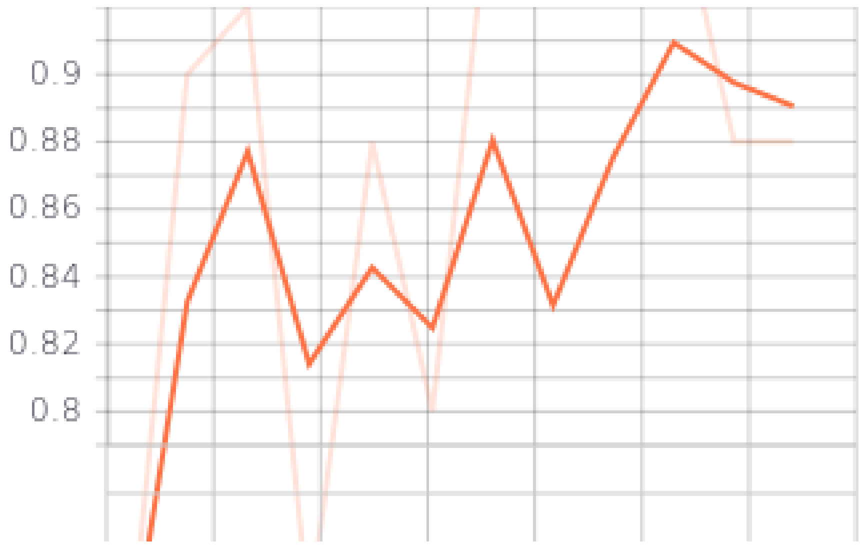


Figure 1: Accuracy curve

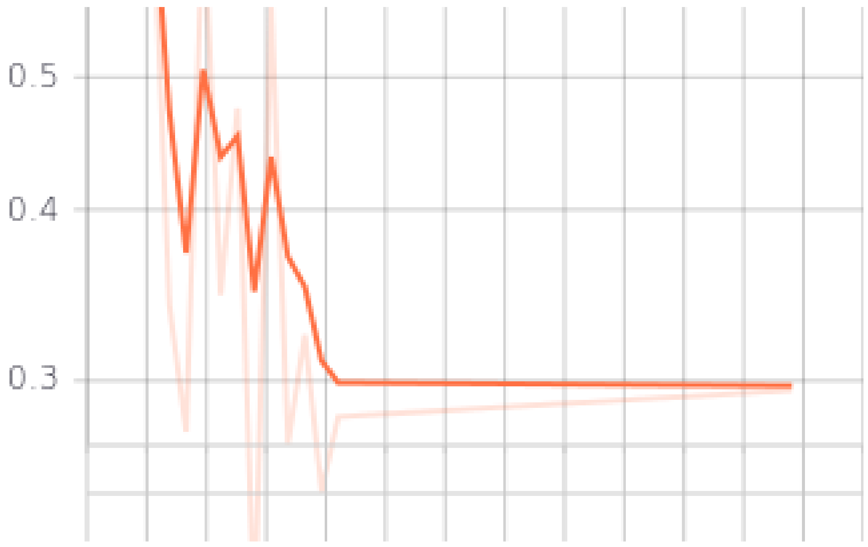


Figure 2: Loss curve

3.2.2 Accuracy

Report the overall accuracy and the per-class accuracy:

Overall Accuracy	88 %

Table 5: Overall Accuracy for Convolutional Neural Network

Class	**Accuracy**
T-shirt/top	80%
Trouser	93%
Pullover	80%
Dress	91%
Coat	92%
Sandal	97%
Shirt	66%
Sneaker	92%
Bag	97%
Ankle boot	97%

Table 6: Per Class Accuracy for Convolutional Neural Network

Identify the problematic classes and list the possible reasons as to why these classes may have significantly lower accuracy compared to other classes.

The most problematic classes were shirt, pullover, and t-shirt/top. Clearly these three classes had low accuracy because the CNN has difficulty distinguishing between them- these three classes are visually very similar and could be difficult to properly separate even for a human.

4 Multiclass Adaboost and Support Vector Machines

4.1 Adaboost: Theory

(a) Show that

$$D_{T+1}(i) \;=\; \frac{\frac{1}{m}\, e^{-F_{T,y_i}(x_i)}}{\prod_{t=1}^{T} Z_t}\,.$$

By evaluating the recursion definition of $D_{T+1}(i)$, (and given that $D_1(i) = \frac{1}{m}$), we know that:

$$D_{T+1}(i) \;=\; \frac{1}{m}\prod_{t=1}^{T}(\frac{e^{-\alpha_t \tilde{h}_{t,y_i}(x_i)}}{Z_t}) \;=\; \frac{\frac{1}{m}\prod_{t=1}^{T} e^{-\alpha_t \tilde{h}_{t,y_i}(x_i)}}{\prod_{t=1}^{T} Z_t} \;=\; \frac{\frac{1}{m} e^{\sum_{t=1}^{T} -\alpha_t \tilde{h}_{t,y_i}(x_i)}}{\prod_{t=1}^{T} Z_t} \;=\; \frac{\frac{1}{m} e^{-F_{T,y_i}(x_i)}}{\prod_{t=1}^{T} Z_t}$$

(b) Show that

$$(H(x_i) \neq y_i) \;\leq\; \big(F_{T,y_i}(x_i) < 0\big)\,.$$

When $H(x_i) = y_i$, we have $1(H(x_i) \neq y_i) = 0$ and the statement is trivially true. When, $H(x_i) \neq y_i$, then $\exists j$ such that $H(x_i) = y_j$. Then, $F_{T,y_i}(x_i)$ is subtracted from every time a weak classifier classifies it as y_j, (which must be happening more than it being classified as y_i because $H(x_i) = y_j$) ultimately this will cause $F_{T,y_i}(x_i) < 0$. So, the statement still holds true.

(c) Show that

$$_S[H] \;\leq\; \frac{1}{m}\sum_{i=1}^{m} e^{-F_{T,y_i}(x_i)} \;=\; \prod_{t=1}^{T} Z_t\,.$$

Hint: For the inequality, use the result of part (b) above, and the fact that $(u < 0) \leq e^{-u}$; for the equality, use the result of part (a) above.

$$ER_s[H] = \frac{1}{m}\sum_{i=1}^{m} 1(H(x_i) \neq y_i) \leq \frac{1}{m}\sum_{i=1}^{m} 1(F_{T,y_i}(x_i) < 0)$$

using part (b)

$$\leq \frac{1}{m}\sum_{i=1}^{m} e^{-F_{T,y_i}(x_i)}$$

from the hint

$$= D_{T+1}(i)\prod_{t=1}^{T} Z_t$$

from part(a)

$$= \prod_{t=1}^{T} Z_t$$

(d) Show that for the given choice of α_t, we have

$$Z_t \;=\; 2\sqrt{_t(1-_t)}\,.$$

(e) Suppose $_t \leq -\gamma$ for all t (where $0 < \gamma \leq$). Then show that

$$_S[H] \;\leq\; e^{-2T\gamma^2}.$$

$$er_s[H] = \prod_{t=1}^{T} Z_t = \prod_{t=1}^{T} 2\sqrt{r_t(1-r_t)} = \prod_{t=1}^{T}(1-2\gamma = (1-2\gamma)^T$$

If we plot $e^{-2\gamma^2}$ it is greater than $(1-2\gamma)$ $\forall \gamma \in (0, .5)$.
So,

$$(1-2\gamma) \leq e^{-2\gamma^2}$$

So,

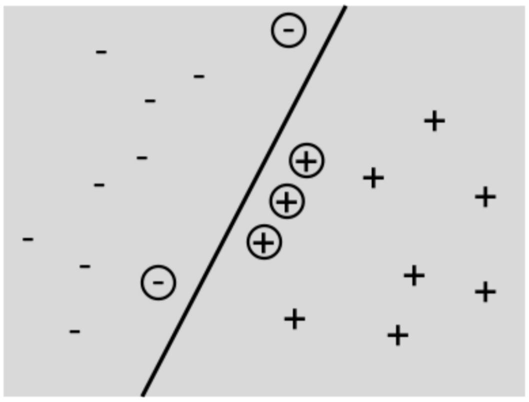
$$(1-2\gamma)^T \leq e^{-2T\gamma^2}$$

This means that

$$er_s[H] \leq e^{-2T\gamma^2}$$

.

4.2 Support Vector Machine



What is the largest number of data points that can be removed from the training set without changing the hard margin SVM solution? Explain your solution.

You can remove 17 data points: any of the non-support vectors, as well as 2 positive support vectors. This is because the two negative support vectors fix the rotational position of the decision boundary. So, the positive support vectors just fix the positioning of the boundary between the positive and negative. Because the positive support vectors are in a line that is parallel to the decision boundary, only one of these vectors is needed to fix this positioning.