Sharif University of Technology
Computer Engineering Department
Ali Sharifi-Zarchi

Homework Assignment 3
Machine Learning for Bioinformatics,
Spring 2023

1401-12-11

Hossein Kargar, Maryam Fazeli

1 Short Questions

1.1

You're training a neural network and notice that the validation error is significantly lower than the training error. Name two possible reasons for this to happen and suggest two possible solutions.

Solution :

The model performs better on unseen data than on training data - this should not happen under normal circumstances.

Possible explanations:

- Training and Validation data sets are not from the same distribution.
- Error in the implementation

1.2

Why do the layers in a deep architecture need to be non-linear?

Solution :

Without nonlinear activation functions, each layer simply performs a linear mapping of the input to the output of the layer. Because linear functions are closed under composition, this is equivalent to having a single (linear) layer. Thus, no matter how many such layers exist, the network can only learn linear functions.

1.3

You are searching for the best learning rate (α) for your model. You decide to test the following values between 0.01 and 1:

$$\alpha \in \{0.01, 0.16, 0.21, 0.84, 0.94\}$$

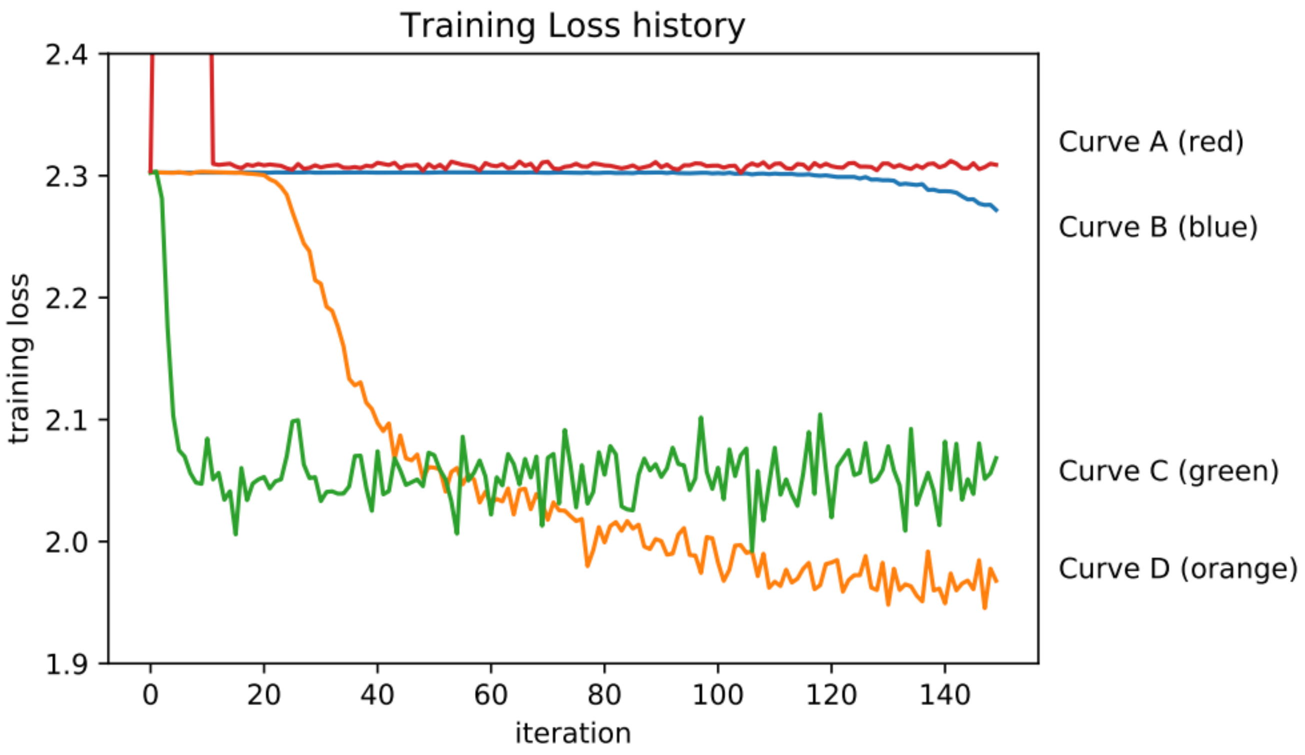
Is that a good method? Explain why.

Solution :

No. A better method would be to use random search on a log scale. Here, we search four values between 0.1 and 1, and search only one between 0.01 and 0.1.

1.4

Your colleague trained a neural network using standard stochastic gradient descent and L2 weight regularization with four different learning rates (α) and plotted the corresponding loss curves. Unfortunately, he forgot which curve belongs to which learning rate. Please assign each of the learning rate values below to the curve (A/B/C/D) it probably belongs to and explain your thoughts. $\alpha = [3e4, 4e1, 2e5, 8e3]$



Solution :

Curve A: $4e-1 = 0.4$ (Learning Rate is way too high)
Curve B: $2e-5 = 0.00002$ (Learning Rate is too low)
Curve C: $8e-3 = 0.008$ (Learning Rate is too high)
Curve D: $3e-4 = 0.0003$ (Good Learning Rate)

1.5

Which of the following is true about the vanishing gradient problem?
(i) Tanh is usually preferred over sigmoid because it doesn't suffer from vanishing gradients.
(ii) Vanishing gradient causes deeper layers to learn more slowly than earlier layers.
(iii) Leaky ReLU is less likely to suffer from vanishing gradients than sigmoid.
(iv) None of the above.

Solution : (iii) is true.

1.6

Describe one advantage of using :
(i) mini-batch gradient descent instead of full-batch gradient descent.
Answer : less computationally expensive, faster convergence
(ii) mini-batch gradient descent instead of stochastic gradient descent with batch size 1.
Answer : faster convergence, less divergence, can leverage efficient vectorized libraries
(iii) Adam optimizer instead of vanilla gradient descent.
Answer : per-parameter updates lead to faster convergence, momentum helps avoid getting stuck in saddle point

2 Backpropagation

The softmax function has the desirable property that it outputs a probability distribution, and is often used as an activation function in many classification neural networks. Consider a 2-layer neural network for K-class classification using softmax activation and cross-entropy loss, as defined below:

$$z^{[1]} = W^{[1]}x + b^{[1]}$$
$$a^{[1]} = LeakyReLU(z^{[1]}, \alpha = 0.01)$$
$$z^{[2]} = W^{[2]}x + b^{[2]}$$
$$\widehat{y} = softmax(z^{[2]})$$
$$L = -\sum_{i=1}^{K} y_i log(\widehat{y_i})$$

where the model is given input x of shape $D_x \times 1$, and one-hot encoded label $y \in \{0,1\}^K$. Assume that the hidden layer has D_a nodes, i.e. $z^{[1]}$ is a vector of size $D_a \times 1$. Recall, the softmax function is computed as follows:

$$\widehat{y} = [\frac{exp(z_1^{[2]})}{Z}, ..., \frac{exp(z_K^{[2]})}{Z}]^T$$

where $Z = \sum_{j=1}^{K} exp(z_j^{[2]})$

2.1

What are the shapes of $w^{[2]}, b^{[2]}$? If we were vectorizing across m examples, i.e. using a batch of samples $X \in \mathbb{R}^{D_x \times m}$ as input, what would be the shape of the output of the hidden layer?

Solution :

$$\mathrm{W}^{[2]} \in \mathbb{R}^{K \times D_a}, \quad b^{[2]} \in \mathbb{R}^{K \times 1} \text{ The output has size } \mathrm{D}_a \times m$$

2.2

What is $\frac{\partial \widehat{y}_k}{\partial z_k^{[2]}}$? Simplify your answer in terms of element(s) of $\widehat{y}$.

Solution :

$$\frac{\partial \hat{\mathbf{y}}_k}{\partial z_k^{[2]}} = \frac{\exp\left(z_k^{[2]}\right) Z - \exp\left(z_k^{[2]}\right)^2}{Z^2} = \hat{\mathbf{y}}_k \left(1 - \hat{\mathbf{y}}_k\right)$$

2.3

What is $\frac{\partial \widehat{y}_k}{\partial z_i^{[2]}}$, for $i \neq k$? Simplify your answer in terms of element(s) of $\widehat{y}$.

Solution :

$$\frac{\partial \hat{\mathbf{y}}_k}{\partial z_i^{[2]}} = -\frac{\exp\left(z_k^{[2]}\right) \exp\left(z_i^{[2]}\right)}{Z^2} = -\hat{\mathbf{y}}_k \hat{\mathbf{y}}_i$$

2.4

Assume that the label y has 1 at its k^{th} entry, and 0 elsewhere. What is $\frac{\partial L}{\partial z_i^{[2]}}$? Simplify your answer in terms of $\widehat{y}$. (Consider both cases where $i = k$ and $i \neq k$).

Solution :

$$\begin{aligned} L &= -\log\left(\hat{\mathbf{y}}_k\right) \\ \text{When } i &= k \\ \frac{\partial L}{\partial z_i^{[2]}} &= \frac{\partial L}{\partial \hat{\mathbf{y}}_k} \frac{\partial \hat{\mathbf{y}}_k}{\partial z_i^{[2]}} = -\frac{1}{\hat{\mathbf{y}}_i} \hat{\mathbf{y}}_i \left(1 - \hat{\mathbf{y}}_i\right) = \hat{\mathbf{y}}_i - 1 \\ \text{When } i &\neq k \\ \frac{\partial L}{\partial z_i^{[2]}} = \frac{\partial L}{\partial \hat{\mathbf{y}}_k} \frac{\partial \hat{\mathbf{y}}_k}{\partial z_i^{[2]}} &= \frac{1}{\hat{\mathbf{y}}_k} \hat{\mathbf{y}}_k \hat{\mathbf{y}}_i = \hat{\mathbf{y}}_i \end{aligned}$$

2.5

What is $\frac{\partial z^{[2]}}{\partial \alpha^{[1]}}$? Refer to this result as δ_1.

Solution :

$$\delta_1 = W^{[2]}$$

2.6

What is $\frac{\partial \alpha^{[1]}}{\partial z^{[1]}}$? Refer to this result as δ_2.

Solution :

$$\delta_2 = \begin{cases} 1, & \text{if } z_i^{[1]} \geq 0 \\ 0.01, & \text{otherwise} \end{cases}$$

2.7

Denote $\frac{\partial L}{\partial z^{[2]}}$ with δ_0. What is $\frac{\partial L}{\partial W^{[1]}}$ and $\frac{\partial L}{\partial b^{[1]}}$? You can reuse notations from previous parts.

Solution :

Using chain rule for partial derivatives, we get: (Note the element-wise product)

$$\begin{aligned} \partial L/\partial W^{[1]} &= \delta_1^T * \delta_0 \circ \delta_2 * \mathbf{x}^T \\ \partial J/\partial b^{[1]} &= \delta_1^T * \delta_0 \circ \delta_2 \end{aligned}$$

2.8

To avoid running into issues with numerical stability, one trick may be used when implementing the softmax function. Let $m = max_{i=1}^{K} z_i$ be the max of z_i,then

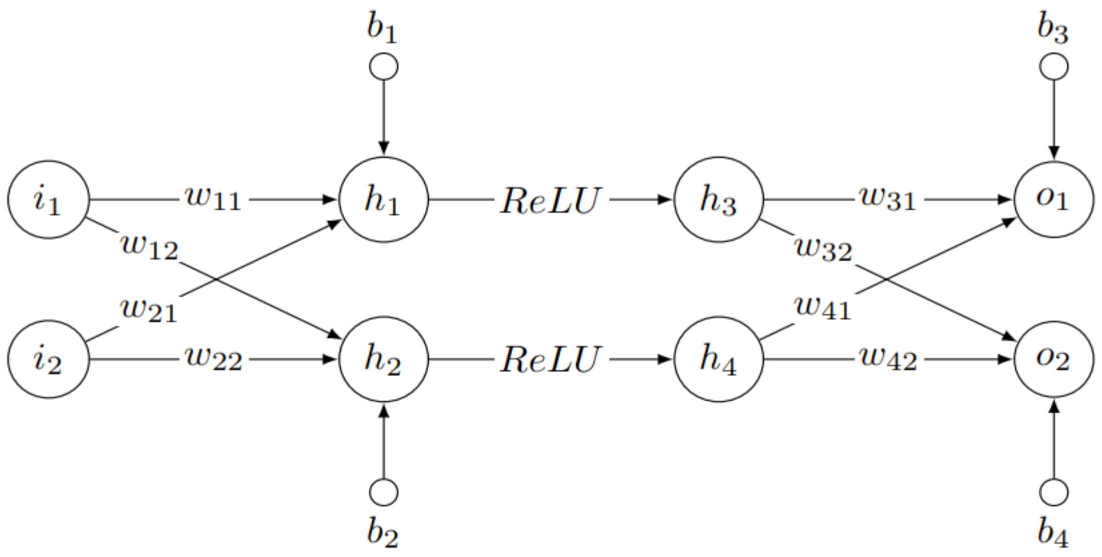
$$\widehat{y}_i = \frac{exp(z_i^{[2]} - m)}{\sum_{j=1}^{K} exp(z_j^{[2]} - m)}$$

What is the numerical problem with the initial softmax computation? Why the modified formula would help to resolve that problem?

> A big $z_i^{[2]}$ will lead to overflow. All very small $z_i^{[2]}$ will lead to division-by-zero error.
> The modified formula ensure the maximum exponential result is 1. and the denominator is always greater than 1.

3 Two Layer Neural Network

Given the following neural network with fully connected layers and ReLU activations, including two input units (i_1, i_2), four hidden units (h_1, h_2) and (h_3, h_4). The output units are indicated as (o_1, o_2) and their targets are indicated as (t_1, t_2). The weights and biases of the fully connected layer are called w and b with specific sub-descriptors.



The values of the variables are given in the following table:

Variable	i_1	i_2	w_{11}	w_{12}	w_{21}	w_{22}	w_{31}	w_{32}	w_{41}	w_{42}	b_1	b_2	b_3	b_4	t_1	t_2
Value	2.0	-1.0	1.0	-0.5	0.5	-1.0	0.5	-1.0	-0.5	1.0	0.5	-0.5	-1.0	0.5	1.0	0.5

3.1

Compute the output (o_1, o_2) with the input (i_1, i_2) and network parameters as specified above. Write down all calculations, including intermediate layer results.

Forward pass:

$$\begin{aligned}
h_1 &= i_1 \times w_{11} + i_2 \times w_{21} + b_1 = 2.0 \times 1.0 - 1.0 \times 0.5 + 0.5 = 2.0 \\
h_2 &= i_1 \times w_{12} + i_2 \times w_{22} + b_2 = 2.0 \times -0.5 + -1.0 \times -1.0 - 0.5 = -0.5 \\
h_3 &= \max(0, h_1) = h_1 = 2 \\
h_4 &= \max(0, h_2) = 0 \\
o_1 &= h_3 \times w_{31} + h_4 \times w_{41} + b_3 = 2 \times 0.5 + 0 \times -0.5 - 1.0 = 0 \\
o_2 &= h_3 \times w_{32} + h_4 \times w_{42} + b_4 = 2 \times -1.0 + 0 \times 1.0 + 0.5 = -1.5
\end{aligned}$$

3.2

Compute the mean squared error of the output (o_1, o_2) calculated above and the target (t_1, t_2).

$$MSE = \frac{1}{2} \times (t_1 - o_1)^2 + \frac{1}{2} \times (t_2 - o_2)^2 = 0.5 \times 1.0 + 0.5 \times 4.0 = 2.5$$

3.3

Update the weight w_{21} using gradient descent with a learning rate of 0.1 as well as the loss computed previously. (Please write down all your computations.)

Backward pass (Applying chain rule):

$$\begin{aligned}
\frac{\partial MSE}{\partial w_{21}} &= \frac{\partial \frac{1}{2}(t_1 - o_1)^2}{\partial o_1} \times \frac{\partial o_1}{\partial h_3} \times \frac{\partial h_3}{\partial h_1} \times \frac{\partial h_1}{\partial w_{21}} + \frac{\partial \frac{1}{2}(t_2 - o_2)^2}{\partial o_2} \times \frac{\partial o_2}{\partial h_3} \times \frac{\partial h_3}{\partial h_1} \times \frac{\partial h_1}{\partial w_{21}} \\
&= (o_1 - t_1) \times w_{31} \times 1.0 \times i_2 + (o_2 - t_2) \times w_{32} \times 1.0 \times i_2 \\
&= (0 - 1.0) \times 0.5 \times -1.0 + (-1.5 - 0.5) \times -1.0 \times -1.0 \\
&= 0.5 + -2.0 = -1.5
\end{aligned}$$

Update using gradient descent:

$$w_{21}^{+} = w_{21} - lr * \frac{\partial MSE}{\partial w_{21}} = 0.5 - 0.1 * -1.5 = 0.65$$