DL-HW1

Jianing Zhang

Jan 30th 2024

1.1 Two-Layer Neural Nets

1.1.1 Regression Task

Answer to (a)

The 5 programming steps that would take to train this model are :

1. Generate the prediction through the model: $\tilde{\mathbf{y}} = model(\mathbf{x})$.
2. Evaluate the cost between $\mathbf{y}$ and $\tilde{\mathbf{y}}$: $L(\mathbf{w}, \mathbf{x}, \mathbf{y}) = F(\mathbf{x}, \mathbf{y}) = C(\mathbf{y}, \tilde{\mathbf{y}}) = \|\tilde{\mathbf{y}} - \mathbf{y}\|^2$.
3. Zero the gradient parameter: $\nabla_{\mathbf{w}} L = 0$
4. Compute and accumulate the gradient parameters: $\nabla_{\mathbf{w}} L = \frac{\partial}{\partial \mathbf{w}} C(\mathbf{y}, \tilde{\mathbf{y}}) = 2(\tilde{\mathbf{y}} - \mathbf{y}) \cdot \frac{\partial \tilde{\mathbf{y}}}{\partial \mathbf{w}}$
5. Step in-towards (to update the parameters): $\mathbf{w}_{\text{new}} = \mathbf{w} - \eta \nabla_W L$

Answer to (b)

For a single data point (x, y),
Layer 1:
Linear layer 1:INPUT is: $\boldsymbol{x}$. OUTPUT is: $s_1 = \mathbf{W}^{(\mathbf{1})} \cdot \boldsymbol{x} + \mathbf{b}^{(\mathbf{1})}$.
function f: INPUT is :$\mathbf{W}^{(\mathbf{1})} \cdot \boldsymbol{x} + \mathbf{b}^{(\mathbf{1})}$, OUTPUT is: $a_1 = f(s_1) = 3ReLu(\mathbf{W}^{(\mathbf{1})} \cdot \boldsymbol{x} + \mathbf{b}^{(\mathbf{1})})$.
Layer 2:
Linear layer 2: INPUT is: $f(s_1) = 3\text{ReLU}(\mathbf{W}^{(\mathbf{1})} \cdot \boldsymbol{x} + \mathbf{b}^{(\mathbf{1})})$, OUTPUT is: $s_2 = \mathbf{W}^{(\mathbf{2})} \cdot a_1 + \mathbf{b}^{(\mathbf{2})} = \mathbf{W}^{(\mathbf{2})} \cdot (3\text{ReLU}(\mathbf{W}^{(\mathbf{1})} \cdot \boldsymbol{x} + \mathbf{b}^{(\mathbf{1})})) + \mathbf{b}^{(\mathbf{2})}$.
Function g: INPUT is: $\mathbf{W}^{(\mathbf{2})} \cdot (3\text{ReLU}(\mathbf{W}^{(\mathbf{1})} \cdot \boldsymbol{x} + \mathbf{b}^{(\mathbf{1})})) + \mathbf{b}^{(\mathbf{2})}$, OUTPUT is:$\tilde{\mathbf{y}} = g(s_2) = s_2 = \mathbf{W}^{(\mathbf{2})} \cdot a_1 + \mathbf{b}^{(\mathbf{2})} = \mathbf{W}^{(\mathbf{2})} \cdot (3\text{ReLU}(\mathbf{W}^{(\mathbf{1})} \cdot \boldsymbol{x} + \mathbf{b}^{(\mathbf{1})})) + \mathbf{b}^{(\mathbf{2})}$.

Answer to (c)

The gradients calculated from the backward pass would be:

$$\begin{aligned}
\frac{\partial C}{\partial \mathbf{W^{(2)}}} &= \frac{\partial C}{\partial \tilde{\mathbf{y}}} \cdot \frac{\partial \tilde{\mathbf{y}}}{\partial \mathbf{s_2}} \cdot \frac{\partial \mathbf{s_2}}{\partial \mathbf{W^{(2)}}} \\
&= \frac{\partial C}{\partial \tilde{\mathbf{y}}} \cdot \frac{\partial \tilde{\mathbf{y}}}{\partial \mathbf{s_2}} \cdot \mathbf{a_1} \\
&= \frac{\partial C}{\partial \tilde{\mathbf{y}}} \cdot \frac{\partial \tilde{\mathbf{y}}}{\partial \mathbf{s_2}} \cdot 3ReLU(\mathbf{W^{(1)}}\boldsymbol{x} + \mathbf{b^{(1)}}) \\
\frac{\partial C}{\partial \mathbf{b^{(2)}}} &= \frac{\partial C}{\partial \tilde{\mathbf{y}}} \cdot \frac{\partial \tilde{\mathbf{y}}}{\partial \mathbf{s_2}} \cdot \frac{\partial \mathbf{s_2}}{\partial \mathbf{b^{(2)}}} \\
&= \frac{\partial C}{\partial \tilde{\mathbf{y}}} \cdot \frac{\partial \tilde{\mathbf{y}}}{\partial \mathbf{s_2}} \\
\frac{\partial C}{\partial \mathbf{W^{(1)}}} &= \frac{\partial C}{\partial \tilde{\mathbf{y}}} \cdot \frac{\partial \tilde{\mathbf{y}}}{\partial \mathbf{s_2}} \cdot \frac{\partial \mathbf{s_2}}{\partial \mathbf{a_1}} \cdot \frac{\partial \mathbf{a_1}}{\partial \mathbf{s_1}} \cdot \frac{\partial \mathbf{s_1}}{\partial \mathbf{W^{(1)}}} \\
&= \frac{\partial C}{\partial \tilde{\mathbf{y}}} \cdot \frac{\partial \tilde{\mathbf{y}}}{\partial \mathbf{s_2}} \cdot \mathbf{W^{(2)}} \cdot \frac{\partial \mathbf{a_1}}{\partial \mathbf{s_1}} \cdot \boldsymbol{x} \\
\frac{\partial C}{\partial \mathbf{b^{(1)}}} &= \frac{\partial C}{\partial \tilde{\mathbf{y}}} \cdot \frac{\partial \tilde{\mathbf{y}}}{\partial \mathbf{s_2}} \cdot \frac{\partial \mathbf{s_2}}{\partial \mathbf{a_1}} \cdot \frac{\partial \mathbf{a_1}}{\partial \mathbf{s_1}} \cdot \frac{\partial \mathbf{s_1}}{\partial \mathbf{b^{(1)}}} \\
&= \frac{\partial C}{\partial \tilde{\mathbf{y}}} \cdot \frac{\partial \tilde{\mathbf{y}}}{\partial \mathbf{s_2}} \cdot \mathbf{W^{(2)}} \cdot \frac{\partial \mathbf{a_1}}{\partial \mathbf{s_1}}
\end{aligned}$$

Answer to (d)

The elements of the partial derivatives would be: As function f is f$(\cdot)$ = 3ReLU$(\cdot)$

$$a_1 = \begin{cases} 3s_1 & \text{if } s_1 > 0 \\ 0 & \text{if } s_1 \leq 0 \end{cases}$$

Then, we can derive that

$$\frac{\partial a_1}{\partial s_1} = \begin{cases} 3 & \text{if } s_1 > 0 \\ 0 & \text{if } s_1 \leq 0 \end{cases}$$

Therefore, $\frac{\partial \mathbf{a_1}}{\partial \mathbf{s_1}}$ is a matrix where diagonal elements are 3 and others are 0. As g is the identity function, by noticing the matrix,

$$\frac{\partial \tilde{\mathbf{y}}}{\partial \mathbf{s_2}} = I \in \mathbb{R}^{K \times K}$$

Finally, we see $C \in \mathbb{R}$ and $\tilde{y} \in \mathbb{R}^K$. Therefore, $\frac{\partial C}{\partial \tilde{\mathbf{y}}} \in \mathbb{R}^{1 \times K}$.

$$\begin{aligned}
(\frac{\partial C}{\partial \tilde{\mathbf{y}}})_i &= 2(\tilde{\mathbf{y}} - \mathbf{y})_i \\
\frac{\partial C}{\partial \tilde{\mathbf{y}}} &= [2(\tilde{\mathbf{y}}_\mathbf{1} - \mathbf{y_1}), \ldots, 2(\tilde{\mathbf{y}}_\mathbf{K} - \mathbf{y_K})] \in \mathbb{R}^{1 \times K}
\end{aligned}$$

1.1.2 Classification Task

Answer to (a)

For (b), As the function f changes to $f = \tanh$ and function g changes to $g = \sigma$, in the forward pass equation, the OUTPUT of f changes to,

$$\begin{aligned}\mathbf{a_1} &= f(s_1)\\ &= \tanh(s_1)\\ &= \tanh(\boldsymbol{W^{(1)}x} + \boldsymbol{b^{(1)}})\end{aligned}$$

Therefore, the input for Linear Layer 2 is

$$tanh(\boldsymbol{W^{(1)}x} + \boldsymbol{b^{(1)}})$$

and the output of Linear Layer 2 becomes:

$$\boldsymbol{W^{(2)}}tanh(\boldsymbol{W^{(1)}x} + \boldsymbol{b^{(1)}}) + \boldsymbol{b^{(2)}}$$

And as $g = \sigma$, therefore, the input becomes

$$\boldsymbol{W^{(2)}}tanh(\boldsymbol{W^{(1)}x} + \boldsymbol{b^{(1)}}) + \boldsymbol{b^{(2)}}$$

And the output becomes,

$$\begin{aligned}\boldsymbol{\tilde{y}} &= g(s_2)\\ &= \sigma(s_2)\\ &= 1 + exp(-(\boldsymbol{W^{(2)}}tanh(\boldsymbol{W^{(1)}x} + \boldsymbol{b^{(1)}}) + \boldsymbol{b^{(2)}}))^{-1}\end{aligned}$$

For (c), The expression in (c) in the previous question does not change. However, as the functions f and g change, the value of back-propagation changes.

$$\begin{aligned}\frac{\partial C}{\partial \mathbf{W^{(2)}}} &= \frac{\partial C}{\partial \tilde{\mathbf{y}}} \cdot \frac{\partial \tilde{\mathbf{y}}}{\partial \mathbf{s_2}} \cdot \frac{\partial \mathbf{s_2}}{\partial \mathbf{W^{(2)}}}\\ &= \frac{\partial C}{\partial \tilde{\mathbf{y}}} \cdot \frac{\partial \tilde{\mathbf{y}}}{\partial \mathbf{s_2}} \cdot \mathbf{a_1}\\ &= \frac{\partial C}{\partial \tilde{\mathbf{y}}} \cdot \frac{\partial \tilde{\mathbf{y}}}{\partial \mathbf{s_2}} \cdot tanh(\boldsymbol{W^{(1)}x} + \boldsymbol{b^{(1)}})\end{aligned}$$

And the other three have same expression.

For (d),

$$\left(\frac{\partial \boldsymbol{a_1}}{\partial \boldsymbol{s_1}}\right)_{ii} = 1 - \tanh^2(\boldsymbol{s_1})_i$$

Therefore, $\frac{\partial \boldsymbol{a_1}}{\partial \boldsymbol{s_1}}$ is a matrix with the diagonal elements mentioned above, and the others are 0.

$$\left(\frac{\partial \boldsymbol{\tilde{y}}}{\partial \boldsymbol{s_2}}\right)_{ii} = \sigma(\boldsymbol{s_{2i}}) \cdot (1 - \sigma(\boldsymbol{s_{2i}}))$$

and the other elements in the matrix are all 0.

Answer to (b)

As the loss function changes to BCE, so the loss is computed as $D_{\text{BCE}}(y, \hat{y}) = -\frac{1}{K}\sum_{i=1}^{K}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)]$. For (b), the expression is the same as it is expressed in Question 1.1.2(a).

For (c), the expression in (c) is also the same as it was in Question 1.1.1, however, the values change.

For (d),

$$\left(\frac{\partial \boldsymbol{a_1}}{\partial \boldsymbol{s_1}}\right)_{ii} = 1 - \tanh^2(\boldsymbol{s_1})_i$$

Therefore, $\frac{\partial \boldsymbol{a_1}}{\partial \boldsymbol{s_1}}$ is a matrix with the diagonal elements mentioned above and the others are 0.

$$\left(\frac{\partial \tilde{\boldsymbol{y}}}{\partial \boldsymbol{s_2}}\right)_{ii} = \sigma(\boldsymbol{s_{2i}}) \cdot (1 - \sigma(\boldsymbol{s_{2i}}))$$

and the other elements in the matrix are all 0.

$$\left(\frac{\partial C}{\partial \tilde{\boldsymbol{y}}}\right)_i = \frac{1}{K}\left(\frac{\boldsymbol{y_i} - \tilde{\boldsymbol{y}}_{\boldsymbol{i}}}{\tilde{\boldsymbol{y}}_{\boldsymbol{i}}(\tilde{\boldsymbol{y}}_{\boldsymbol{i}} - 1)}\right)$$

Therefore, $\frac{\partial C}{\partial \tilde{\boldsymbol{y}}}$ is a row vector $\in \mathbb{R}^{1\times K}$.

Answer to (c)

ReLU introduces non-linearity, allowing the model to capture complex relationships in data. Its sparsity reduces computational load and overfitting. Moreover, ReLU mitigates vanishing gradient issues, enabling efficient gradient propagation. It facilitates faster convergence, reducing training time. This choice also maintains efficient computation in large-scale networks.

1.2 Conceptual Questions

Answer to (a)

Softmax is often termed ”softargmax” because it transforms the argmax operation into a differentiable, probabilistic function. While argmax picks the maximum value, softmax computes a probability distribution, making the values soft probabilities. It exponentiates and normalizes inputs, enabling gradient-based optimization. This smooth, continuous output is ideal for multi-class classification tasks.

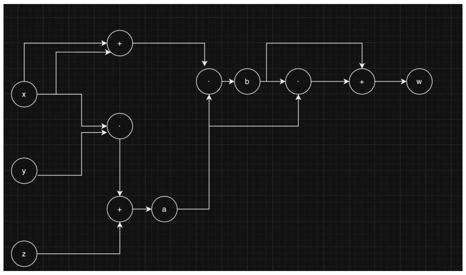
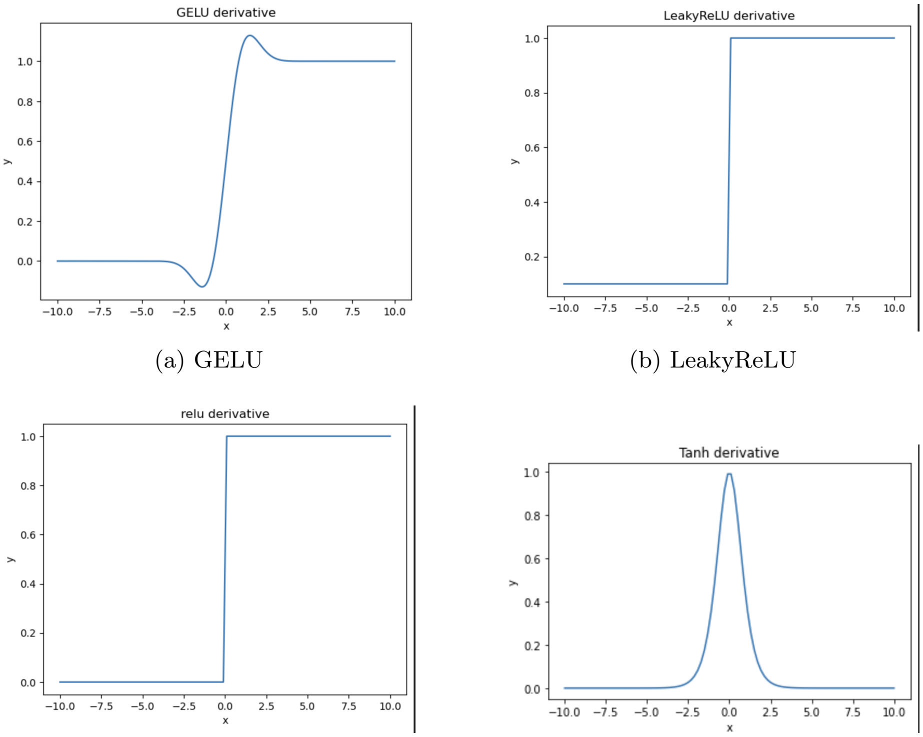


Figure 1: Enter Caption

Answer to (b) is shown above

Answer to (c)



(a) GELU

(b) LeakyReLU

(c) ReLU

(d) Tanh

Figure 2: Activation functions comparison.

Answer to (d)

(a) The Jacobian matrix of f and g are:

$$\text{The Jacobian matrix of } f \text{ is } \mathbf{W}_1 \in \mathbb{R}^{b\times a}$$

$$\text{The Jacobian matrix of } g \text{ is } \mathbf{W}_2 \in \mathbb{R}^{b\times a}$$

(b) The Jacobian matrix of $h(x) = f(x) + g(x)$ is:

$$\mathbf{W}_1 + \mathbf{W}_2 \in \mathbb{R}^{b\times a}$$

(c) The Jacobian matrix of $h(x) = f(x) + g(x)$ if $\mathbf{W}_1 = \mathbf{W}_2$ (so $a = b = c$) is:

$$2\mathbf{W}_1 \text{ or } 2\mathbf{W}_2 \in \mathbb{R}^{a\times a}$$

Answer to (e)

(a) The Jacobian matrix of f and g.

$$\text{The Jacobian matrix of } f \text{ is } \mathbf{W}_1 \in \mathbb{R}^{b\times a}$$

$$\text{The Jacobian matrix of } g \text{ is } \mathbf{W}_2 \in \mathbb{R}^{c\times b}$$

(b) The Jacobian matrix of $h(x) = g(f(x)) = (g \circ f)(x)$ is

$$\begin{aligned}\frac{\partial h}{\partial x} &= \frac{\partial g(f(x))}{\partial x} \\ &= \frac{\partial g}{\partial x} \cdot \frac{\partial f}{\partial x} \\ &= \mathbf{W}_2 \cdot \mathbf{W}_1 \in \mathbb{R}^{c\times a}\end{aligned}$$

(c) The Jacobian matrix of $h(x)$ if $\mathbf{W}_1 = \mathbf{W}_2$ (so $a = b = c$) is

$$\begin{aligned}\frac{\partial h}{\partial x} &= \mathbf{W}_1 \cdot \mathbf{W}_1 \in \mathbb{R}^{a\times a} \\ &= \mathbf{W}_2 \cdot \mathbf{W}_2 \in \mathbb{R}^{c\times c}\end{aligned}$$

1.3 Deriving Loss Functions

The common update rule:

$$w_i \leftarrow w_i + \eta(y - \tilde{y})x_i$$

Perceptron

Perceptron:

$$\tilde{y} = \text{sign}(b + \sum_{i=1}^{d} w_i x_i)$$

We need to find a hyperplane that separates the data successfully by using Perceptron. And this leads to the binary prediction of either 1 or -1, which depends on the side of the hyperplane.
And the updating rule is that we keep w_i same if $\tilde{y} = y$ and we move w_i in the correct direction if $\tilde{y} \neq y$.Though the activation function is binary, we still want our prediction to be more accurate, therefore, there is a $b + \sum_{i=1}^{d} w_i x_i$ in the equation so that we can find that the loss function should be the function of the distance between the prediction and the true value.
As

$$w_i \leftarrow w_i + \eta(y - \tilde{y})x_i$$

we can know that

$$\nabla_c(w_i) = (y - \tilde{y})x_i$$

And all these conditions can be achieved by the loss function

$$Loss(x, y, w) = \nabla_c(w_i) = -(y - \tilde{y}) \cdot \sum_{i=1}^{d} w_i x_i$$

Adaline / Least Mean Squares

Adaline / Least Mean Squares: As a regression problem, Adaline uses an identity function as activation function.

$$\tilde{y} = b + \sum_{i=1}^{d} w_i x_i$$

And we can see w and b as the best linear function mapping x_i to y. And we can use the minimum Euclidean distance between the prediction and the true value as the best case. Therefore, the loss function can be thought as

$$Loss(x, y, w) = \frac{1}{2}(y - \tilde{y})^2 = \frac{1}{2}(y - \sum_{i=1}^{d} w_i x_i)^2$$

The factor of $\frac{1}{2}$ is used for convenience in the derivative calculation as it cancels the exponent when the derivative is taken.

Logistic Regression

Logistic Regression:

$$\tilde{y} = tanh(b + \sum_{i=1}^{d} w_i x_i)$$

This activation function make our output between 1 and -1. Therefore, the loss function should obey the condition that if $y = \tilde{y}$, the loss function is 0. And if $y, \tilde{y}$ diverges, the loss function increases. Therefore, the loss function should be

$$Loss(x, y, w) = -2log(1 + exp(-y \sum_{i=1}^{d} w_i x_i)))$$

This function is convex and has unique solution for w.