

Homework 1

Guilherme Albertini

September 23, 2022

Theory

Let $Linear_1 \rightarrow f \rightarrow Linear_2 \rightarrow g$ be a two-layer neural net architecture whereby $Linear_i(x) = \mathbf{W}^{(i)}\mathbf{x} + \mathbf{b}^{(i)}$ is the i^{th} affine transformation and f, g are element-wise nonlinear activation functions (else must use transposes and/or Hadamard products). When an input $\mathbf{x} \in \mathbb{R}^n$ is fed into the network, $\hat{\mathbf{y}} \in \mathbb{R}^K$ is obtained as output.

Problem 1: Regression Task

We would like to perform regression task. We choose $f(\cdot) = 5(\cdot)^+ = 5ReLU(\cdot)$ and g to be the identity function. To train, we choose MSE loss function, $\ell_{MSE}(\hat{\mathbf{y}}, \mathbf{y}) = ||(\hat{\mathbf{y}} - \mathbf{y})||^2$.

1. Name and mathematically describe the 5 programming steps you would take to train this model with PyTorch using SGD on a single batch of data.

(a) see vid

2. For a single data point (x, y) , write down all inputs and outputs for forward pass of each layer. You can only use variables and mechanics specified prior in your answer.

— $Linear_1$ —
Input: \mathbf{x}
Output (z_1): $\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}$
— f —
Input: $Linear_1(\mathbf{x}) = \mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}$
Output (z_2): $f(Linear_1(\mathbf{x})) = 5ReLU(Linear_1(\mathbf{x}))$
 $= 5ReLU(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)})$
 $= 5\max(0, \mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)})$
— $Linear_2$ —
Input: $f(Linear_1(\mathbf{x}))$
Output (z_3): $Linear_2(f(Linear_1(\mathbf{x}))) = \mathbf{W}^{(2)}f(Linear_1(\mathbf{x})) + \mathbf{b}^{(2)}$
 $= 5\mathbf{W}^{(2)}((\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)})) + \mathbf{b}^{(2)}$
— g —
Input: $Linear_2(f(Linear_1(\mathbf{x})))$
Output: $g(Linear_2(f(Linear_1(\mathbf{x})))) =$
 $5(\mathbf{W}^{(2)}(\max(0, \mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)})) + \mathbf{b}^{(2)})\mathbf{I} =$
 $5(\mathbf{W}^{(2)}(\max(0, \mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)})) + \mathbf{b}^{(2)}) = \hat{\mathbf{y}}$
—Loss—
Input: $\hat{\mathbf{y}}$
Output: $\|(5(\mathbf{W}^{(2)}(\max(0, \mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)})) + \mathbf{b}^{(2)}) - \mathbf{y})\|^2$

3. Write down the gradients calculated from the backward pass. You can only use the following variables: $\mathbf{x}, \mathbf{y}, \mathbf{W}^{(1)}, \mathbf{b}^{(1)}, \mathbf{W}^{(2)}, \mathbf{b}^{(2)}, \frac{\partial \ell}{\partial \hat{\mathbf{y}}}, \frac{\partial \ell}{\partial \mathbf{z}_2}, \frac{\partial \ell}{\partial \mathbf{z}_1}, \frac{\partial \ell}{\partial \mathbf{z}_3}$, where $\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \hat{\mathbf{y}}$ are outputs of $Linear_1, f, Linear_2, g$, respectively.

$$\begin{aligned}
\frac{\partial z_{3i}}{\partial W^{(2)}} &= \frac{\partial(5W^{(2)} \max(0, W^{(1)}x + b^{(1)}) + b^{(2)})}{\partial W^{(2)}} \\
&= 5 \max(0, W^{(1)}x + b^{(1)})^T, \text{ else } 0^T \\
\frac{\partial z_3}{\partial b^{(2)}} &= \frac{\partial(5W^{(2)} \max(0, W^{(1)}x + b^{(1)}) + b^{(2)})}{\partial b^{(2)}} \\
&= \text{diag}(1) \\
\frac{\partial \ell}{\partial W^{(2)}} &= \frac{\partial \ell}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z_3} \frac{\partial z_3}{\partial W^{(2)}} \\
&= 5 \frac{\partial \ell}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z_3} \max(0, W^{(1)}x + b^{(1)})^T \\
\frac{\partial \ell}{\partial b^{(2)}} &= \frac{\partial \ell}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z_3} \frac{\partial z_3}{\partial b^{(2)}} \\
&= \frac{\partial \ell}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z_3}
\end{aligned}$$

$$\begin{aligned}
\frac{\partial z_3}{\partial z_2} &= \frac{\partial(W^{(2)} 5 \max(0, W^{(1)}x + b^{(1)}))}{\partial 5 \max(0, W^{(1)}x + b^{(1)})} = W^{(2)} \\
\frac{\partial z_{1i}}{\partial W^{(1)}} &= \frac{\partial(W^{(1)}x + b^{(1)})}{\partial W^{(1)}} = x^T, \text{ else } 0^T \\
\frac{\partial z_1}{\partial b^{(1)}} &= \frac{\partial(W^{(1)}x + b^{(1)})}{\partial b^{(1)}} = \text{diag}(1) \\
\frac{\partial \ell}{\partial W^{(1)}} &= \frac{\partial \ell}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z_3} \frac{\partial z_3}{\partial z_2} \frac{\partial z_2}{\partial z_1} \frac{\partial z_1}{\partial W^{(1)}} \\
&= \frac{\partial \ell}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z_3} W^{(2)} \frac{\partial z_2}{\partial z_1} x^T \\
\frac{\partial \ell}{\partial b^{(1)}} &= \frac{\partial \ell}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z_3} \frac{\partial z_3}{\partial z_2} \frac{\partial z_2}{\partial z_1} \frac{\partial z_1}{\partial b^{(1)}} \\
&= \frac{\partial \ell}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z_3} W^{(2)} \frac{\partial z_2}{\partial z_1}
\end{aligned}$$

4. Show the elements of $\frac{\partial z_2}{\partial z_1}$, $\frac{\partial \hat{y}}{\partial z_3}$, $\frac{\partial \ell}{\partial \hat{y}}$. Be careful about dimensionality.

Note: $i \in \{1, \dots, m\}$ used throughout.

$$\frac{\partial \mathbf{z}_2}{\partial \mathbf{z}_1} = \frac{\partial(5 \max(0, \mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}))}{\partial(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)})}$$

$$\left(\frac{\partial \mathbf{z}_2}{\partial \mathbf{z}_1}\right)_{ii} = \begin{cases} 0, & z_{1i} < 0 \\ 5, & z_{1i} > 0 \\ \text{undefined (or assigned a value 0 in code)}, & z_{1i} = 0 \end{cases}$$

Note: $\mathbf{z}_1 = \mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}$ and the above is a diagonal matrix

$$\left(\frac{\partial \hat{\mathbf{y}}}{\partial \mathbf{z}_3}\right)_{ii} = 1 \text{ (and 0 elsewhere, off of diagonal; an identity matrix)}$$

$$\frac{\partial \ell}{\partial \hat{\mathbf{y}}} = \frac{\partial(\|\hat{\mathbf{y}} - \mathbf{y}\|^2)}{\partial \hat{\mathbf{y}}} = 2(\hat{\mathbf{y}} - \mathbf{y})^T \text{ (A vector)}$$

Problem 2: Classification Task

We would like to perform multi-class classification task, so we set $f = \tanh$ and $g = \sigma$, the logistic sigmoid function, $\sigma(z) = \frac{1}{1+\exp(-z)}$.

1. If you want to train this network, what do you need to change in the equations of (1.2), (1.3) and (1.4), assuming we are using the same MSE loss function.

— f —

Input: $Linear_1(\mathbf{x}) = \mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}$

Output (z_2): $f(Linear_1(\mathbf{x})) = \tanh(Linear_1(\mathbf{x}))$

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

$= \frac{\exp(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}) - \exp(-\mathbf{W}^{(1)}\mathbf{x} - \mathbf{b}^{(1)})}{\exp(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}) + \exp(-\mathbf{W}^{(1)}\mathbf{x} - \mathbf{b}^{(1)})}$

— $Linear_2$ —

Input: $f(Linear_1(\mathbf{x})) = \tanh(Linear_1(\mathbf{x}))$

Output (z_3): $Linear_2(f(Linear_1(\mathbf{x}))) = \mathbf{W}^{(2)}f(Linear_1(\mathbf{x})) + \mathbf{b}^{(2)}$

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

— g —

Input: $Linear_2(f(Linear_1(\mathbf{x})))$

Output: $g(Linear_2(f(Linear_1(\mathbf{x})))) =$

$\frac{1}{1+\exp(-z_3)} = \frac{1}{1+\exp(-(\mathbf{W}^{(2)} \tanh(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}) + \mathbf{b}^{(2)}))} = \hat{\mathbf{y}}$

—Loss—

Input: $\hat{\mathbf{y}}$

Output: $\|(\frac{1}{1+\exp(-(\mathbf{W}^{(2)} \tanh(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}) + \mathbf{b}^{(2)}))} - \mathbf{y})\|^2$

$$\begin{aligned}
\frac{\partial z_{3i}}{\partial \mathbf{W}^{(2)}} &= \frac{\partial (\mathbf{W}^{(2)} \tanh(\mathbf{W}^{(1)} \mathbf{x} + \mathbf{b}^{(1)}) + \mathbf{b}^{(2)})}{\partial \mathbf{W}^{(2)}} \\
&= \tanh(\mathbf{W}^{(1)} \mathbf{x} + \mathbf{b}^{(1)})^T, \text{ else } 0^T \\
\frac{\partial z_3}{\partial \mathbf{b}^{(2)}} &= \frac{\partial (\mathbf{W}^{(2)} \tanh(\mathbf{W}^{(1)} \mathbf{x} + \mathbf{b}^{(1)}) + \mathbf{b}^{(2)})}{\partial \mathbf{b}^{(2)}} \\
&= \text{diag}(1) \\
\frac{\partial \ell}{\partial \mathbf{W}^{(2)}} &= \frac{\partial \ell}{\partial \hat{\mathbf{y}}} \frac{\partial \hat{\mathbf{y}}}{\partial z_3} \frac{\partial z_3}{\partial \mathbf{W}^{(2)}} \\
&= \frac{\partial \ell}{\partial \hat{\mathbf{y}}} \frac{\partial \hat{\mathbf{y}}}{\partial z_3} \tanh(\mathbf{W}^{(1)} \mathbf{x} + \mathbf{b}^{(1)})^T \\
\frac{\partial \ell}{\partial \mathbf{b}^{(2)}} &= \frac{\partial \ell}{\partial \hat{\mathbf{y}}} \frac{\partial \hat{\mathbf{y}}}{\partial z_3} \frac{\partial z_3}{\partial \mathbf{b}^{(2)}} \\
&= \frac{\partial \ell}{\partial \hat{\mathbf{y}}} \frac{\partial \hat{\mathbf{y}}}{\partial z_3}
\end{aligned}$$

$$\begin{aligned}
\frac{\partial z_3}{\partial \mathbf{z}_2} &= \frac{\partial (\mathbf{W}^{(2)} \tanh(\mathbf{W}^{(1)} \mathbf{x} + \mathbf{b}^{(1)}) + \mathbf{b}^{(2)})}{\partial \tanh(\mathbf{W}^{(1)} \mathbf{x} + \mathbf{b}^{(1)})} = \mathbf{W}^{(2)} \\
\left(\frac{\partial \mathbf{z}_2}{\partial \mathbf{z}_1} \right)_{ii} &= \frac{\partial (\tanh(\mathbf{W}^{(1)} \mathbf{x} + \mathbf{b}^{(1)}))}{\partial (\mathbf{W}^{(1)} \mathbf{x} + \mathbf{b}^{(1)})} = 1 - \tanh(\mathbf{W}^{(1)} \mathbf{x} + \mathbf{b}^{(1)})^2 \\
\left(\frac{\partial \hat{\mathbf{y}}}{\partial z_3} \right)_{ii} &= \frac{\partial (1 + \exp(-(\mathbf{W}^{(2)} \tanh(\mathbf{W}^{(1)} \mathbf{x} + \mathbf{b}^{(1)}) + \mathbf{b}^{(2)})))^{-1}}{\partial (\tanh(\mathbf{W}^{(1)} \mathbf{x} + \mathbf{b}^{(1)}) + \mathbf{b}^{(2)})} \\
&= \sigma(z_3)(1 - \sigma(z_3))_i, \\
&\text{Note the two partials above represent diagonal matrices} \\
\frac{\partial \ell}{\partial \hat{\mathbf{y}}} &= \frac{\partial (\|\hat{\mathbf{y}} - \mathbf{y}\|^2)}{\partial \hat{\mathbf{y}}} = 2(\hat{\mathbf{y}} - \mathbf{y})^T \text{ (A vector)}
\end{aligned}$$

2. Now you think you can do a better job by using a Binary Cross Entropy (BCE) loss function $\ell_{BCE}(\hat{\mathbf{y}}, \mathbf{y}) = \frac{1}{K} \sum_{i=1}^K -[y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)]$. What needs to change from the previous?

—Loss—

Input: $\hat{\mathbf{y}}$

Output: $\ell_{BCE}(\hat{\mathbf{y}}, \mathbf{y}) = \frac{1}{K} \sum_{i=1}^K -[y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)]$,

$$\hat{y}_i = \frac{1}{1 + \exp(-z_{3i})}$$

$$\frac{\partial \ell}{\partial \hat{\mathbf{y}}} = \left(\frac{-y_i}{\hat{y}_i} + \frac{1-y_i}{1-\hat{y}_i} \right)^T$$

3. Things are getting better. You realize that not all intermediate hidden activations need to be binary (or soft version of binary). You decide to use $f(\cdot) = (\cdot)^+$ but keep g as σ . Explain why this choice of f can be beneficial for training a (deeper) network.

Sigmoid function is more computationally intensive to compute compared to ReLU due to exponential operation; the latter produces sparsity in matrices which encourages numerical optimization techniques taking advantage of this time and space complexity reduction.

ReLU also avoids the vanishing gradient problem, whereby the number of parameters receive very small updates such that the nodes deviate greatly from their optimal value. As the gradient is constant for ReLU compared to the sigmoid gradient always being smaller than 1, successive operations will start to prohibit learning.

Problem 3: Conceptual Questions

1. Why is softmax actually softargmax?

Section 2.2

Problem 6

Blah

Problem 7

Blah

Problem 10

Blah