Assignment 3

September 28, 2023

Eleanore who is DYING

Problem 4

Demonstrating the equivalence between a multiple layer neural network without an activation function and a layer of linear networ

Solution 4

Let's firstly consider a 2 layer neural network, while W_1 is weight matrix and b is bias, it would compute $W_1 x + b_1$.

A second layer would then compute:

$$W_2(W_1 x + b_1) + b_2 = W_2 W_1 x + W_2 b_1 + b_2$$

which is equivalent to $\boldsymbol{W'x + b'}$. Also, adding layers will not change the resulte.

Thus, we can concludes that MLP without activation function is equivalent to only a layer of linear network. This also tells us the function of activation function: add non-linear properties.

Problem 5

What does the negative sign signify in Gradient Descent?

Solution 5

GD moves the vector in the **opposite direction** of the current slope towards the minima.

Problem 6

What could be the outcome if there are too many layers with sigmoid as the activation function?

Solution 6

Firstly, since σ is based on exponetial function, the **calculated amount is big**.

Secondly, when we use GD, the fomular for updating weight is,

$$w_{i+1} = w_i - \eta \frac{\partial \mathcal{L}}{\partial w_t}$$

while

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial w_i} &= \frac{\partial \mathcal{L}}{\partial x_i} \cdot \frac{\partial x_i}{\partial z_i} \cdot \frac{\partial z_i}{\partial w_i} \\
&= \frac{\partial \mathcal{L}}{\partial x_i} \cdot \sigma'(z_i) x_{i-1}
\end{aligned}$$

Since the derivative of σ is

$$\sigma'(z) = \frac{e^{-z}}{(1 + e^{-z})^2} = \sigma(z)(1 - \sigma(z))$$

Also, the range of derivative of σ is $(0, 0.25)$.

Thus, in the process of BP, as we approaching input layer, the continued multiplication will become smaller, causing **the update of gradient become slower**. In this situation, the neural network just work in shallow layers, in fact.

Problem 7

Prove $\tanh(z) + 1 = 2\sigma(2z)$, and explore their potential relationship and why we replace sigmoid with tanh. (hint: range, derivative)

Solution 7

As we know,

$$\tanh(z) = \frac{1 - e^{-2z}}{1 + e^{-2z}}$$
$$\tanh(z) + 1 = \frac{2}{1 + e^{-2z}}$$

while

$$2\sigma(2z) = 2\frac{1}{1 + e^{-2z}}$$

Thus, $\tanh(z) + 1 = 2\sigma(2z)$

Then,let's look at the difference between this two function by graph.

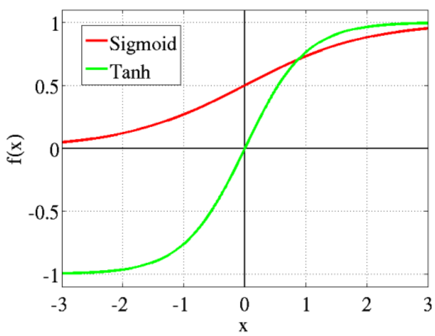

Figure 1: the image of tanh vs. sigmoid

Transformation from σ to tanh make the center(inflection point) of activation function change from 0.5 to 0. Thus, **use of tanh will make the probability distribution after activating centered at 0** rather than 0.5, which is more natural.

Then, let's find the derivatives.

$$\sigma'(z) = \sigma(z)(1 - \sigma(z))$$

$$\begin{aligned} \tanh'(z) &= \frac{4e^{-2z}}{(1 + e^{-2z})^2} \\ &= \frac{(1 + e^{-2z})^2 - (1 - e^{-2z})^2}{(1 + e^{-2z})^2} \\ &= 1 - \tanh(z)^2 \end{aligned}$$

Calculating and comparing the range of derivative for tanh and σ, we find,

$$\tanh'(z) : (0, 0.25)$$
$$\sigma'(z) : (0, 1)$$

Thus, **larger derivatives of tanh lead to faster convergence during training**, as updates to the model's parameters are more substantial.

Problem 8

How can the problem of Overfitting be solved? Provide a list of at least three methods and illustrate two of them.

Solution 8

1. Imporve training dataset. We could have find or create more data.

2. Randonly dropout some point in training set We could randomly egnore some of the neuro in the process of training.

3. Use simple model rather than complicated one.

Problem 9

Thinking: Why does model training require more VRAM than inference? Not necessary to prove it, show me your guess.

Solution 9

In the process of **training**, which is usually refers to BP. It requires space to **store each weight's gradients and learning rates**.

Inference refers to FP, where only the parameters of network need to be active in the memory. The activations are discarded once the forward pass moves to a new layer. Hence, only the layer that is active in memory and the layer that gets calculated are comsumpting memory. Thus, inference only needs to continuously **hold the network parameters and temporarily hold two feature maps**.