Assignment$_5$

Chaitra , Ganesh , Avik

June 7 2019

1 Question 1

$$1.f(x) = cos(x)^2, g(x) = 2e^x, h(x) = x^2 - 2, r(x) = 1/x^3$$

Find

$$1.\delta(f(g(h(r(x))))/\delta(x)$$

Answer:

$$r(x) = 1/x^3$$

$$h(r(x)) = 1/x^6 - 2$$

$$g(h(r(x))) = 2\mathrm{e}^{\frac{1}{x^6}-2}$$

$$f(g(h(r(x)))) = cos(2\mathrm{e}^{\frac{1}{x^6}-2})^2$$

$$f'(x) = -sin(2\mathrm{e}^{\frac{1}{x^6}-2})^2.4\mathrm{e}^{\frac{1}{x^6}-2}.\mathrm{e}^{\frac{1}{x^6}}. - 6\frac{1}{x^7}$$

$$f'(x) = 48sin(2\mathrm{e}^{\frac{1}{x^6}-2})^2.\mathrm{e}^{\frac{1}{x^6}-2}/x^7$$

$$2.\delta(h(r(g(f(x^2 - 2xy + y^2)))))/\delta(y)$$

Answer:

$$f(x) = cos(x - y)^4$$

$$g(f(x)) = 2\mathrm{e}^{cos(x-y)^4}$$

$$r(g(x)) = 1/(2\mathrm{e}^{cos(x-y)^4})^3$$

$$h(r(g(x))) = (1/(2\mathrm{e}^{cos(x-y)^4})^3)^2 - 2$$

$$h(r(g(x))) = (1/64\mathrm{e}^{cos(x-y)^{24}}) - 2$$

$$h'(y) = 1/64(\mathrm{e}^{-cos(x-y)^{24}}.sin(x - y)^{24}.24(x - y)^{23}(-1))$$

$$h'(y) = -3(\mathrm{e}^{-cos(x-y)^{24}}.sin(x - y)^{24}.(x - y)^{23})/8$$

2 Question 2

In this exercise you will see that training and artificial neural network (ANN) is very similar to training a SVM. Our ANN will have four layers: an input layer, two hidden layers of size k each and one output layer. A scheme can be seen in Figure 1a.

a) Find,

$$a^4 -> f, W^2, W^3, W^4, x, a^2, a^3$$

Answer:

$$a^2 = f(W^2.x)$$

$$a^3 = f(W^3.a^2)$$

$$a^4 = f(W^4.a^3) = f(W^4.f(W^3.a^2)) = f(W^4.f(W^3.f(W^2.x)))$$

b) Find derivative of the sigmoid function.

Answer:

$$f(x) = 1/1 + e^{-x}$$

$$f'(x) = -1(1 + e^{-x})^{-2}e^{-x}(-1) = (1 + e^{-x})^{-2}e^{-x}$$

$$= e^{-x}/(1 + e^{-x})^2 = 1 + e^{-x} - 1/(1 + e^{-x})^2$$

$$= ((1 + e^{-x})/(1 + e^{-x})^2) - (1/(1 + e^{-x})^2)$$

$$= (1/(1 + e^{-x})) - (1/(1 + e^{-x})^2)$$

$$= 1/(1 + e^{-x})(1 - (1/1 + e^{-x}))$$

c) Find,

$$1.\delta J(\theta)/\delta W_t^4$$

Answer:

$$\delta J(\theta)/\delta W^4 = \delta J(\theta)/\delta a^4 * \delta a^4/\delta W^4$$

$$= \sum_{t=1}^{n}(a_4 - y_t) * \delta\sigma(W_t^4.a_t^3)/\delta W_t^4$$

$$= \sum_{t=1}^{n}(a_4 - y_t) * \sigma(W_t^4.a_t^3) * (1 - \sigma(W_t^4.a_t^3)) * a_t^3$$

$$2.\delta J(\theta)/\delta W^3$$

Answer:

$$\delta J(\theta)/\delta W^3 = \delta J(\theta)/\delta a^4 * \delta a^4/\delta a^3 * \delta a^3/\delta W^3$$

$$= \sum_{t=1}^{n}(a_4 - y_t) * \delta\sigma(W_t^4.a_t^3)/\delta a_t^3 * \delta\sigma(W_t^3.a_t^2)/\delta W_t^3$$

$$= \sum_{t=1}^{n}(a_4 - y_t) * \sigma(W_t^4.a_t^3) * (1 - \sigma(W_t^4.a_t^3)) * W_t^4 * \sigma(W_t^3.a_t^2) * (1 - \sigma(W_t^3.a_t^2)) * a_t^2$$

$$3.\delta J(\theta)/\delta W^2$$

Answer:

$$\delta J(\theta)/\delta W^2 = \delta J(\theta)/\delta a^4 * \delta a^4/\delta a^3 * \delta a^3/\delta a^2 * \delta a^2/\delta W^2$$

$$= \sum_{t=1}^{n}(a_4 - y_t) * \delta\sigma(W_t^4.a_t^3)/\delta a_t^3 * \delta\sigma(W_t^3.a_t^2)/\delta a_t^2 * \delta\sigma(W_t^2.a_t^1)/\delta W_t^2$$

$$= \sum_{t=1}^{n}(a_4 - y_t) * \sigma(W_t^4.a_t^3) * (1 - \sigma(W_t^4.a_t^3)) * W_t^4 * \sigma(W_t^3.a_t^2) * (1 - \sigma(W_t^3.a_t^2)) * W_t^3 * \sigma(W_t^2.a_t^1) * (1 - \sigma(W_t^2.a_t^1)) * a_t^1$$

d) Explain how you could use the stochastic gradient descent to find the optimal values of the weights. You can do it writing a small text or doing pseudocode.
Answer: Stochastic Gradient Descent updates parameters in an iterative manner to optimize the loss function. Instead of iterating through all training samples for the cost function, it updates the weight parameters after iterating through every training sample which is chosen at random.

for one or more epochs, until the approximate cost minimum is reached:
for training sample i:
for each weight j :

$$w_j := w + \Delta w_j, where : \Delta w_j = n(target^i - output^i)x_j^i$$

3 Question 3

1. The aim of this question is show the similarities between the SVM and ANN optmizations.
a) Show that

$$lim_{(a^{**} - a^*) -> -\infty} ln(e^{a1} + e^{a2} + ... + e^{an}) = max(a1, a2, ..., an) = a^*$$

Answer: Here, a* is the first maximum number and a** is the second maximum number.

Let d be the greatest difference between a** and a*, a*=n, then Left Hand Side will be reduced to below since we can ignore other terms (smaller values)

$$lim_{d->\infty} ln(e^n + e^{n-d}) = lim_{d->\infty} ln.e^n(1 + e^{-d})$$
$$= n * lim_{d->\infty} ln(1 + e^{-d})$$

when

$$d- > \infty, e^{-d} - > 0$$

$$= n + lim_{d->\infty} ln(1)$$

$$= n + ln(1)$$

$$= n = a^* = max(a1, a2, ..., an)$$

b) Using item (a), show that the SVM hinge loss can be approximated asymptotically by the calculated limit.
Answer:
SVM Hinge Loss = max (0, 1-t)

a* = 1-t and a** = 0

This can be expressed as,

$$lim_{1-t->-\infty}(ln(e^{1-t}) + e^0) = lim_{1-t->-\infty} ln(1 + e^{1-t})$$

This is how we approximated SVM asymptotically by applying limit (1-t) $\rightarrow -\infty$

Therefore,

$$h^*(t) = ln(1 + e^{1-t})$$

c) Compare

$$h(t) \rightarrow l(t) = ln(1 + e^{-t})$$

Answer:

By comparing

$$log(1 + e^{1-t}) ① \rightarrow log(1 + e^{-t}) ②$$

with t=+5, we get,

$$log(1 + e^{-4}) \approx 0.0181499 ①$$

$$log(1 + e^{-5}) \approx 0.0067153 ②$$

The loss of ② is lower than the approximated ①