pen_paper_stienstra

Arent Stienstra 10074279

November 12, 2017

1 Exercise 1

$$\frac{\partial \mathcal{L}}{\partial W_{out}} = (y_{out} - y_{gt}) \cdot f_3'(W_{out} \cdot f_2 \cdot (W_2 \cdot f_1(W_1 x_{in}) \cdot f_2 \cdot (W_2 \cdot f_1(W_1 x_{in}) \tag{1}$$

$$\frac{\partial \mathcal{L}}{\partial W_{out}} = (y_{out} - y_{gt}) \cdot f_3'(s_{out}) \cdot z_2 \tag{2}$$

$$\frac{\partial \mathcal{L}}{\partial W_2} = (y_{out} - y_{gt}) \cdot f_3'(s_{out}) \cdot W_{out} \cdot f_2'(s_1) \cdot z_1 \tag{3}$$

$$\frac{\partial \mathcal{L}}{\partial W_1} = (y_{out} - y_{gt}) \cdot f_3'(s_{out}) \cdot W_{out} \cdot f_2'(s_2) \cdot W_2 \cdot f_1'(s_1) \cdot x_{in} \tag{4}$$

1.1 Error propagation

$$\frac{\partial \mathcal{L}}{\partial W_k} = \delta k \cdot z_{k-1} \tag{5}$$

2 Exercise 2

2.1 Feed forward

$$\text{s1} = X^t \dot{W}_1 = \begin{matrix} 0.458 & 0.869 & 0.704 \\ 0.121 & 0.1615 & 0.044 \\ -0.442 & -0.181 & 0.704 \\ 0.120 & 0.119 & -0.044 \end{matrix} \quad \text{z1} = \text{relu(s1)} = \begin{matrix} 0.458 & 0.869 & 0.704 \\ 0.1205 & 0.1615 & 0.044 \\ 0 & 0 & 0.704 \\ 0.1195 & 0.1185 & 0 \end{matrix}$$

$$\text{s2= out= } = z1^t \dot{W}_2 = \begin{matrix} 0.099 \\ 0.011 \\ 0.049 \\ 0.002 \end{matrix} \text{ loss} = 3.89479953265$$

2.2 Back propagation

$$\frac{\partial \mathcal{L}}{\partial S_2} == S_2 - Y = \begin{matrix} -0.901 \\ -0.989 \\ 1.049 \\ 1.002 \end{matrix}$$

$$\frac{\partial out}{\partial W_2} == z_1^T \cdot$$

$$\partial \mathcal{L} \partial out = \begin{matrix} -0.412 \\ -0.824 \\ 0.060 \end{matrix}$$

$$DS1 = \frac{\partial out}{\partial S_1} = \frac{\partial \mathcal{L}}{\partial out} \dot{W_2}^T \odot relu_{der}(S_1) = \begin{matrix} -0.018 & -0.027 & -0.081 \\ -0.020 & -0.030 & -0.089 \\ 0.000 & 0.000 & 0.094 \\ 0.020 & 0.030 & 0.000 \end{matrix}$$

$$\frac{\partial S_1}{\partial W_1} = x^T \dot{D} S1 = \begin{matrix} 0.013 & -0.020 & -0.149 \\ -0.013 & -0.020 & -0.149 \\ -0.016 & -0.025 & 0.006 \end{matrix}$$

2.3 Weight updates

$$W_2 = W_2 - 0.5 * D_W 2 = \begin{matrix} 0.226 \\ 0.442 \\ 0.060 \end{matrix}$$

$$W_1 = W_1 - 0.5 * D_W 1 = \begin{matrix} 0.607 & 0.710 & 0.075 \\ 0.018 & 0.442 & 0.877 \end{matrix}$$

3 Exercise 3

3.1 general

The Hinge function has a linear penalty for the additional certainty for classes that are not correct compared to the certainty for the class that is correct. The certainty is measured with the softmax.

3.2 derivative

$$\mathcal{L}_{hinge}{}^i = \sum_{j \neq y_i} max(\frac{e^{O_j} - e^{O_{y_i}}}{\sum_k e^{O_k}}) \tag{6}$$

Case $O_j > O_{Y_i}$

$$\frac{\partial \mathcal{L}_{hinge}{}^i}{\partial O_j} = \frac{e^{O_j}(1 + e^{O_{y_i}} - e^{O_j})}{(\sum_k e^{O_k})^2} \tag{7}$$

Case $O_j <= O_{Y_i}$

$$\frac{\partial \mathcal{L}_{hinge}{}^{i}}{\partial O_j} = 0 \tag{8}$$

4 Exercise 4

The current algorithm saves the output and activation of each hidden layer so the back propagation can use those results to calculate the gradients w.r.t. the weights and biases.

One could also only save the outputs of each layer and then recalculate the activation's when doing back propagation. Especially with activation functions such as the RELU, where the computational effort to calculate the activation's is very low, this could save a lot of memory without sacrificing too much speed.

In general, one should be aware of the computational effort the activation's take and how much memory saving the activation's cost.