TTIC 31020: Introduction to Machine Learning
Problem Set 8

Hung Le Tran

28 Feb 2024

Problem 8.1 (Back Propagation)

(a) 1. Sigmoid case. Denote $\sigma =$ sigmoid. Then we have that

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

Then we fix some v whose activation function was $\sigma =$ sigmoid. Then

$$a[v] = \sum_{(u,v)\in E} w(u,v)o[u],$$

which is the weighted output of the signals from the parent nodes of v then

$$o[v] = \sigma\left(\sum_{(u,v)\in E} w(u,v)o[u]\right)$$

We can first break down

$$\frac{\partial \hat{y}}{\partial w(u,v)} = \frac{\partial \hat{y}}{\partial a[v]}\frac{\partial a[v]}{\partial w(u,v)} = \frac{\partial \hat{y}}{\partial a[v]}o[u]$$

So it remains for us to calculate $\frac{\partial \hat{y}}{\partial a[v]}$. Denote this $\delta[v]$.

We suppose that we have calculated $\delta[v_{out}]$, and that our concerned $v \neq v_{out}$. Then since its $a[v]$ contributes to $\hat{y}$ via its children nodes, we get:

$$\begin{aligned}\delta[v] = \frac{\partial \hat{y}}{a[v]} &= \sum_{(v,w)\in E} \frac{\partial \hat{y}}{\partial a[w]}\frac{\partial a[w]}{\partial o[v]}\frac{\partial o[v]}{\partial a[v]} \\ &= \sum_{(v,w)\in E} \delta[w]w(v,w)\sigma'(a[v])\end{aligned}$$

So

$$\frac{\partial \hat{y}}{w(u,v)} = \delta[v]o[u] = \left(\sum_{(v,w)\in E} \delta[w]\sigma'(a[v])\right)o[u]$$

2. Softmax case. Our activation is now

$$a[v] = \left(\sum_{u\in Parent(v)} w(1,u,v)o[u], \dots, \sum_{u\in Parent(v)} w(k,u,v)o[u]\right)^T \in \mathbb{R}^k, \quad o[v] = \text{softmax}(a[v])$$

In particular for an index $i \in [k]$, we write out explicitly

$$a[v]_i = \sum_{u\in Parent(v)} w(i,u,v)o[u]$$

Since $\text{softmax}(z_1, \ldots, z_n) = \frac{\sum_j z_j e^{z_j}}{\sum_j e^{z_j}}$, we get

$$\begin{aligned}
\frac{\partial \text{softmax}(z_1, \ldots, z_n)}{\partial z_i} &= \frac{(\sum_j e^{z_j})(z_i e^{z_i} + e^{z_i}) - (\sum_j z_j e^{z_j})(e^{z_i})}{(\sum_j e^{z_j})^2} \\
&= \frac{e^{z_i}(1 + z_i - \text{softmax}(z_1, \ldots, z_n))}{\sum_j e^{z_j}}
\end{aligned}$$

For sake of conciseness, call this $\text{softmax}_i(z_1, \ldots, z_n)$ where we keep in mind that the sub-i is not an index, but rather a partial.

Fix i, u, v. Then, we have

$$\frac{\partial \hat{y}}{\partial w(i,u,v)} = \frac{\partial \hat{y}}{\partial a[v]_i} \frac{\partial a[v]_i}{\partial w(i,u,v)} = \frac{\partial \hat{y}}{\partial a[v]_i} o[u]$$

Vectorize this, then

$$\left(\frac{\partial \hat{y}}{\partial w(i,u,v)} \right)_{u \in Parent(v)} = \frac{\partial \hat{y}}{a[v]} o[u]$$

So we try to calculate the stimulus

$$\delta[v] = \frac{\partial \hat{y}}{\partial a[v]} \in \mathbb{R}^k$$

Then, with $k(w) = \{v' : (v', w) \in E\}$, we have

$$\begin{aligned}
\delta[v]_i &= \frac{\partial \hat{y}}{\partial a[v]_i} \\
&= \sum_{(v,w) \in E} \sum_{j=1}^{k(w)} \frac{\partial \hat{y}}{\partial a[w]_j} \frac{\partial a[w]_j}{\partial o[v]} \frac{\partial o[v]}{\partial a[v]_i} \\
&= \sum_{(v,w) \in E} \sum_{j=1}^{k(w)} \delta[w]_j w(j,v,w) \text{softmax}_i(a[v])
\end{aligned}$$

because $o[v]$ contributes to all indices j of the activation $a[w]$ for each w with weight $w(j,v,w)$. We thus established the recurrence relationship.

3. For both procedures, we have not filled in the gap of the calculation of $\delta[v_{out}]$. If v_{out} uses a sigmoid activation function, then

$$\begin{aligned}
\delta[v_{out}] &= \frac{\partial \hat{y}}{a[v_{out}]} \\
&= \frac{\partial o[v_{out}]}{\partial a[v_{out}]} = \sigma'(a[v_{out}])
\end{aligned}$$

Otherwise, if v_{out} uses a softmax activation function, then

$$\begin{aligned}
\delta[v_{out}] &= \frac{\partial \text{softmax}(a[v_{out}])}{a[v_{out}]} \\
&= [\text{softmax}_1(a[v_{out}]), \text{softmax}_2(a[v_{out}]), \ldots, \text{softmax}_{k(v_{out})}(a[v_{out}])]^T \\
&= \nabla \text{softmax}(a[v_{out}])
\end{aligned}$$

is the standard derivative of the softmax. We've thus tied up all loose ends.

(b) First we perform the forward propagation:

$$\begin{aligned} x &= o[0] \in \mathbb{R}^d \\ a[1] &= W^{(1)} o[0] \in \mathbb{R}^k \\ o[1] &= \text{sigmoid}(a[1]) \in \mathbb{R}^k \\ a[2] &= W^{(2)} o[1] \in \mathbb{R}^k \\ \hat{y} = o[2] &= \text{softmax}(a[2]) \in \mathbb{R} \end{aligned}$$

We have that

$$\frac{\partial \ell^{sq}(\hat{y}(x), y)}{\partial \hat{y}} = (\hat{y} - y)$$

Thus we are now only concerned with

$$\nabla_{W^{(2)}} \hat{y} \text{ and } \nabla_{W^{(1)}} \hat{y}$$

1. For $W^{(2)}$. Consider $W^{(2)}_{i,j}$ with $i, j \leq k$. Then

$$\begin{aligned} \frac{\partial \hat{y}}{\partial W^{(2)}_{i,j}} &= \frac{\partial \hat{y}}{\partial a[2]_i} \frac{\partial a[2]_i}{\partial W^{(2)}_{i,j}} \\ &= \text{softmax}_i(a[2]) o[1]_j \end{aligned}$$

so in matrix form, it is the outer product of ∇softmax$(a[2])$ and $o[1]$:

$$\nabla_{W^{(2)}} \hat{y} = (\nabla \text{softmax}(a[2]))\, o[1]^T$$

So

$$\nabla_{W^{(2)}} \ell^{sq}(\hat{,} y) = (\hat{y} - y)\, (\nabla \text{softmax}(a[2]))\, o[1]^T$$

2. For $W^{(1)}$.

We know that

$$\delta^{(2)} = \frac{\partial \hat{y}}{\partial a[2]} = \nabla \text{softmax}(a[2]) \in \mathbb{R}^k$$

From above analysis, we have

$$\delta^{(1)} = \frac{\partial \hat{y}}{\partial a[1]} = \text{sigmoid}'(a^{(1)}) \odot {W^{(2)}}^T \delta^{(2)} \in \mathbb{R}^k$$

So

$$\begin{aligned} \frac{\partial \hat{y}}{\partial W^{(1)}_{i,j}} &= \delta^{(1)}_i \frac{\partial a[1]_i}{\partial W^{(1)}_{i,j}} = \delta^{(1)}_i o[0]_j \\ \Rightarrow \nabla_{W^{(1)}} \hat{y} &= \delta^{(1)} o[0]^T \\ \Rightarrow \nabla_{W^{(1)}} \ell^{sq}(\hat{y}, y) &= (\hat{y} - y) \delta^{(1)} o[0]^T \end{aligned}$$

Problem 8.2 (Expressive Power of Neural Networks)

Take some $h_I(x) \in PARITIES_d$, where $I = \{i_1, \ldots, i_K\}$ for some $I \subseteq [d]$. Obviously $K \leq d$. Let $p_I(x) =$ number of 1's of x. Then clearly $h_I(x) = p_I(x) \bmod 2$.

Let us describe the architecture:

$$\begin{aligned} x &= (x_1, \ldots, x_d) \in \{0, 1\}^d \\ a[1]_i &= \langle w_i, x \rangle + b_i \\ o[1] &= ReLU(a[1]) \text{ (element-wise)} \\ a[2] &= \text{sign}\left(\sum_{i=1}^{2d} a_i o[1]_i \right) \end{aligned}$$

Then set $(w_i)_{i_k} = 1$ for all $k \in [K]$ and 0 otherwise. It then follows that $\langle w_i, x \rangle = p_I(x)$.