Problem 5.1

A “spiral network” is a brand new category of neural network, invented just for this homework. It is a network with a scalar input variable $x_i$, a scalar target variable $y_i$, and with the following architecture:

$$h_{i,j} = \begin{cases} x_i, & j = 1 \\ g(\xi_{i,j}), & 2 \leq j \leq M \\ \sum_{k=1}^{j-1} w_{j,k} h_{i,k} & \end{cases}$$

$$\xi_{i,j} = j - 1 \sum_{k=1}^{M} w_{j,k} h_{i,k}$$

Suppose that the network is trained to minimize the sum of the per-token squared errors $E = \frac{1}{2} \sum_{i=1}^{n} (h_{i,M} - y_i)^2$. The error gradient can be written as

$$\frac{\partial E}{\partial w_{j,k}} = \sum_{i=1}^{n} \delta_{i,j} h_{i,k}$$

Find a formula that can be used to compute $\delta_{i,j}$, for all $2 \leq j \leq M$, in terms of $y_i$, $h_{i,j} = g(\xi_{i,j})$, and/or $g'(\xi_{i,j}) = \frac{dg}{d\xi_{i,j}}$.

Problem 5.2

The back-prop of a convolution layer is correlation. What about if correlation is the forward-prop rule? Let’s find out. Consider a “correlational” layer, given as follows, where $h[m_1,m_2]$ is the hidden node activation of the previous layer, and $w[m_1,m_2]$ are the network weights:

$$\xi[n_1,n_2] = w[-n_1,-n_2] * h[n_1,n_2]$$

$$= \sum_{m_1} \sum_{m_2} w[m_1 - n_1, m_2 - n_2] h[m_1,m_2]$$

Suppose the loss, $L$, is some function whose derivatives with respect to $\xi[n_1,n_2]$, $\delta[n_1,n_2] = \frac{dE}{d\xi[n_1,n_2]}$, are known. Find $\frac{dE}{dn[m_1,m_2]}$ and $\frac{dE}{dw[k,\xi]}$ in terms of $\delta[n_1,n_2]$.

Problem 5.3

Consider the following nonlinearity, which might be appropriate at the output layer of a classifier. This nonlinearity is sometimes called the “softcount” nonlinearity, and is closely related to the more common “softmax.” The softmax and softcount share the following property: the input, $\xi$, and output, $h$ are both
Homework 5 2

assumed to be vectors, $\vec{\xi} = [\xi_1, \ldots, \xi_{NY}]^T$ and $\vec{h} = [h_1, \ldots, h_{NY}]^T$. The $k^{\text{th}}$ output of the nonlinearity depends on all of the inputs, not just on the $k^{\text{th}}$ input:

$$h_k = g_k(\vec{\xi}) = \frac{e^{\xi_k}}{\max_{1 \leq \ell \leq NY} e^{\xi_\ell}}$$

Suppose that the training target, $y$, is an integer, $1 \leq y \leq NY$, and the loss is the categorical cross-entropy function:

$$\mathcal{L} = -\sum_{k=1}^{NY} 1[y = k] \ln h_k$$

where

$$1[P] = \begin{cases} 1 & P \text{ is true} \\ 0 & \text{otherwise} \end{cases}$$

Find $\frac{d\mathcal{L}}{d\xi_k}$, for each of the following four cases:

(a) $k = y$ and $k = \arg\max_{\ell} e^{\xi_\ell}$

(b) $k = y$ but $k \neq \arg\max_{\ell} e^{\xi_\ell}$

(c) $k \neq y$ but $k = \arg\max_{\ell} e^{\xi_\ell}$

(d) $k \neq y$ and $k \neq \arg\max_{\ell} e^{\xi_\ell}$

Express your answer in terms of $h_\ell$, for any $1 \leq \ell \leq NY$ including possibly $\ell = k$, $\ell = y$, or $\ell = \arg\max e^{\xi_\ell}$. Do not express your answer in terms of $\xi_k$.

Problem 5.4

Consider a two-layer regression network with $N_x$ input nodes, $N_h$ hidden nodes, and $N_y$ output nodes:

$$\tilde{f}(\vec{x}) = W^{(2)} \sigma \left(W^{(1)} \vec{x}\right)$$

(5.4-1)

Suppose that there are $N_i$ training tokens, $\mathcal{D} = \{ (\vec{x}_1, \vec{y}_i), \ldots, (\vec{x}_{Ni}, \vec{y}_{Ni}) \}$, and the loss is mean-squared error:

$$\mathcal{L} = \frac{1}{N_i} \sum_{i=1}^{N_i} ||\tilde{f}(\vec{x}_i) - \vec{y}_i||_2^2$$

(5.4-2)

- How many multiply-accumulate operations are required to calculate the gradients $\nabla_{W^{(2)}} \mathcal{L}$ and $\nabla_{W^{(2)}} \mathcal{L}$ using forward-propagation and back-propagation?

- Suppose you attempted to find these gradients using a forward-Euler approximation,

$$\frac{\partial \mathcal{L}}{\partial w_{k,j}^{(l)}} \approx \frac{1}{\epsilon} \left( \mathcal{L}(w_{k,j}^{(l)} + \epsilon) - \mathcal{L}(w_{k,j}^{(l)}) \right),$$

(5.4-3)

for some suitably small value of $\epsilon$. How many multiply-accumulate operations would be required to compute $\nabla_{W^{(2)}} \mathcal{L}$ and $\nabla_{W^{(2)}} \mathcal{L}$ using Eq. (5.4-3)?