Assignment One

Matthew C. Scicluna
Département d'Informatique et de Recherche Opérationnelle
Université de Montréal
Montréal, QC H3T 1J4
matthew.scicluna@umontreal.ca

February 11, 2018

1 Neural Networks Classification

We consider training a standard feed-forward neural network using a single iteration of SGD on a single training example (x, y). Let $f(x, \theta)$ as the output of the neural network with model parameters θ. Let g be the output activation function and $a(x, \theta)$ is the pre-activation network output s.t. $f(x, \theta) = g(a(x, \theta))$.

(a) An appropriate activation function for the output layer is the softmax activation function

$$g(a(x,\theta)) = \sigma(a(x,\theta)) = \frac{1}{1+\exp(a(x,\theta))}$$

(b) The output of the softmax activation function represents $p(y = 1|x, \theta)$

(c) We use the definition of L_{CE} from (6.12) of [1] and (b) to get that:

$$\begin{aligned}
L_{CE}(f(x,\theta), y) &= -\mathbb{E}_{(X,Y)\sim\hat{p}}\{\log p(Y|X,\theta)\} \\
&= -\log\left(p(y=1|x,\theta)^y p(y=0|x,\theta)^{(1-y)}\right) \\
&= -\log\left(f(x,\theta)^y (1-f(x,\theta))^{(1-y)}\right) \\
&= -y\log f(x,\theta) - (1-y)\log(1-f(x,\theta))
\end{aligned}$$

(d) We take the derivative of the expression in (c) and use the fact that $\frac{\partial \sigma(x)}{\partial x} = \sigma(x)(1-\sigma(x))$ to get that:

$$\begin{aligned}
\frac{\partial L_{CE}(f(x,\theta),y)}{\partial a(x,\theta)} &= -y\frac{\partial \log\sigma(a(x,\theta))}{\partial a(x,\theta)} - (1-y)\frac{\partial \log(1-\sigma(a(x,\theta))}{\partial a(x,\theta)} \\
&= -y\frac{\sigma(a(x,\theta))(1-\sigma(a(x,\theta)))}{\sigma(a(x,\theta))} + (1-y)\frac{\sigma(a(x,\theta))(1-\sigma(a(x,\theta)))}{1-\sigma(a(x,\theta))} \\
&= -y(1-\sigma(a(x,\theta))) + (1-y)\sigma(a(x,\theta)) \\
&= -(y - f(x,\theta))
\end{aligned}$$

(e) We use the definition of L_{CE} from (6.13) of [1] and (b) to get that:

$$\begin{aligned}
L_{MSE}(f(x,\theta),y) &= \frac{1}{2}\mathbb{E}_{(X,Y)\sim\hat{p}}\left\{\|Y - f(X,\theta)\|^2\right\} \\
&= \frac{1}{2}\|y - f(x,\theta)\|^2
\end{aligned}$$

(f) We take the derivative of the expression in (e) to get:

$$\begin{aligned}
\frac{\partial L_{MSE}(f(x,\theta),y)}{\partial a(x,\theta)} &= (y - f(x,\theta))\frac{\partial f(x,\theta)}{\partial a(x,\theta)} \\
&= (y - f(x,\theta))\frac{\partial f(x,\theta)}{\partial a(x,\theta)} \\
&= -(y - f(x,\theta))f(x,\theta)(1-f(x,\theta))
\end{aligned}$$

(g) For binary classification, the more appropriate loss function would be the Cross Entropy loss function. We can interpret this loss as the negative log likelihood, meaning that minimizing this loss is equivalent to finding the maximimum likelihood estimate. The mean squared loss doesn't have this interpretation in this context, and so it is not clear what we could interpret its minimum as.

2 Neural Network Representation

We consider the binary classification problem described in the figure with classes being represented by solid triangles and empty squares. We denote the squares as Class 0 and the triangles as Class 1. We want to derive a classifier network with a single hidden layer with 3 units with Heaviside step functions H as activations. We let

$$f(x_1,x_2) = H\left(\sum_{j=1}^{3} u_j h_j + c\right) \quad \text{and } h_j = H\left(\sum_{i=1}^{2} w_{ij}x_i + b_j\right)$$

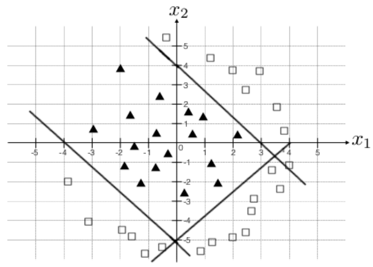

Figure 2.1: Decision boundaries learned from the neural network

From figure 2.1 we see that the class membership can be perfectly determined using three decision boundaries. If our neural network could use the preactivation weights to learn each decision boundary it could use the hidden units to determine class membership. The weights would be as follows:
The topmost decision boundary is $-\frac{4}{3}x_1 - x_2 + 4 = 0$ For positive class membership it is necessary that $-\frac{4}{3}x_1 - x_2 + 4 > 0$. So $w_{11} = \frac{3}{4}$, $w_{21} = -1, b_1 = 4$ will give us that $h_j = 1$ iff the necessary condition is satisfied. Likewise we have $\frac{5}{3}x_1 - x_2 - 5 < 0$ which gives us $w_{12} = -\frac{5}{3}$, $w_{22} = 1, b_2 = 5$. Finally, $-\frac{5}{4}x_1 - x_2 - 5 < 0$ gives us $w_{13} = \frac{5}{4}$, $w_{23} = 1, b_3 = 5$. For membership in class 1, it easy to see that it is both neccessary and sufficient that $h_1 = h_2 = h_3 = 1$. If we set $u_1 = u_2 = u_3 = 1, c = -2.5$ we have that:

$$f(x_1, x_2) = 1 \iff \sum_{j=1}^{3} h_j - 2.5 > 0 \iff h_1 = h_2 = h_3 = 1$$

Giving us the desired result.

3 Activation Functions

(a) We show that the derivative of the $ReLU(x) := \max(0, x)$ (where it exists) is the Heaviside step function:

$$H(x) = \begin{cases} 1 & x > 0 \\ 0 & x < 0 \\ \frac{1}{2} & x = 0 \end{cases}$$

We consider 3 cases. When $x > 0$ we have that:

$$\begin{aligned} \frac{\partial max(x,0)}{\partial x} &= \lim_{\epsilon \to 0} \frac{\max(x+\epsilon, 0) - \max(x, 0)}{\epsilon} \\ &= \lim_{\epsilon \to 0} \frac{x + \epsilon - x}{\epsilon} \\ &= 1 \end{aligned}$$

Where the second line follows when $x > |\epsilon|$. When x<0 we have:

$$\begin{aligned}
\frac{\partial max(x,0)}{\partial x} &= \lim_{\epsilon \to 0} \frac{\max(x+\epsilon,0) - \max(x,0)}{\epsilon} \\
&= \lim_{\epsilon \to 0} \frac{\max(x+\epsilon,0)}{\epsilon} \\
&= 0
\end{aligned}$$

Where, as before, the last line follows when $x > |\epsilon|$. When $x = 0$ the limit does not exist. This is because the one sided limits are not equal:

$$\begin{aligned}
\lim_{\epsilon \to 0^+} \frac{\max(\epsilon,0)}{\epsilon} &= \lim_{\epsilon \to 0^+} \frac{\epsilon}{\epsilon} = 1 \\
\lim_{\epsilon \to 0^-} \frac{\max(\epsilon,0)}{\epsilon} &= 0
\end{aligned}$$

(b) We can define $ReLU$ in terms of H in the following way:

$$\begin{aligned}
ReLU(x) &:= xH(x) \\
ReLU(x) &:= \int_{t=-\infty}^{x} H(t)dt
\end{aligned}$$

(c) We can write $H(x)$ as an asymptotic expression using $\sigma(x)$

$$H(x) = \lim_{\epsilon \to \infty} \sigma(\epsilon x)$$

(d) As in (a) we break this into three pieces. Let $x > 0$ then:

$$\begin{aligned}
\frac{\partial H(x)}{\partial x} &= \lim_{\epsilon \to 0} \frac{H(x+\epsilon) - H(x)}{\epsilon} \\
&= \lim_{\epsilon \to 0} \frac{1-1}{\epsilon} \\
&= 0
\end{aligned}$$

Let $x < 0$:

$$\begin{aligned}
\frac{\partial H(x)}{\partial x} &= \lim_{\epsilon \to 0} \frac{H(x+\epsilon) - H(x)}{\epsilon} \\
& \lim_{\epsilon \to 0} \frac{0-0}{\epsilon} \\
&= 0
\end{aligned}$$

As in (a), the limit does not exist at $x = 0$:

$$\begin{aligned}
\lim_{\epsilon \to 0^+} \frac{H(\epsilon) - \frac{1}{2}}{\epsilon} &= \lim_{\epsilon \to 0^+} \frac{1 - \frac{1}{2}}{\epsilon} = \lim_{\epsilon \to 0^+} \frac{1}{2\epsilon} = \infty \\
\lim_{\epsilon \to 0^-} \frac{H(\epsilon) - \frac{1}{2}}{\epsilon} &= \lim_{\epsilon \to 0^-} \frac{0 - \frac{1}{2}}{\epsilon} = \lim_{\epsilon \to 0^+} \frac{-1}{2\epsilon} = -\infty
\end{aligned}$$

Notice $\frac{\partial H(x)}{\partial x} = 0$ for all $x \neq 0$ and that we can assign any value at $x = 0$. If we let the derivative be ∞ at 0 we have that $\frac{\partial H(x)}{\partial x}$ is the dirac delta function, as needed.

4 Gradients and Networks

(a) We compute the Jacobian of the softmax function $S(x)_i = \frac{\exp(x_i)}{\sum_k \exp(x_k)}$. First we consider the case where $i \neq j$

$$\begin{aligned}
\frac{\partial S(x)_i}{\partial x_j} &= \frac{\partial}{\partial x_j} \frac{\exp(x_i)}{\sum_k \exp(x_k)} \\
&= \exp(x_i) \frac{\partial}{\partial x_j} \left(\sum_k \exp(x_k) \right)^{-1} \\
&= -\exp(x_i) \left(\sum_k \exp(x_k) \right)^{-2} \frac{\partial}{\partial x_j} \sum_k \exp(x_k) \\
&= -\exp(x_i) \left(\sum_k \exp(x_k) \right)^{-2} \exp(x_j) \\
&= -S(x)_i S(x)_j
\end{aligned}$$

When $i = j$ we have that:

$$\begin{aligned}
\frac{\partial S(x)_i}{\partial x_i} &= \frac{\exp(x_i) \left(\sum_k \exp(x_k) \right) - \exp(x_i)^2}{\left(\sum_k \exp(x_k) \right)^2} \\
&= S(x)_i - \frac{\exp(x_i)^2}{\left(\sum_k \exp(x_k) \right)^2} \\
&= S(x)_i - S(x)_i^2 \\
&= S(x)_i (1 - S(x)_i)
\end{aligned}$$

(b) We can express (a) as the following matrix equation:

$$J(x) = Diag(S(x)) - S(x)S(x)^T$$

Where $Diag(S(x))$ is a diagonal matrix whose i^{th} entry is $S(x)_i$.

(c) We compute the jacobian matrix of the logistic sigmoid function, applied element-wise to x. For $i \neq j$ we have that:

$$\frac{\partial \sigma(x)_i}{\partial x_j} = 0$$

For $i = j$ we have:

$$\begin{aligned}
\frac{\partial \sigma(x)_j}{\partial x_j} &= \frac{\partial}{\partial x_j} \frac{\exp(x_j)}{1 + \exp(x_j)} \\
&= \frac{\exp(x_j)(1 + \exp(x_j)) - \exp(x_j)^2}{(1 + \exp(x_j))^2} \\
&= \frac{\exp(x_j)}{1 + \exp(x_j)} - \left(\frac{\exp(x_j)}{1 + \exp(x_j)} \right)^2 \\
&= \sigma(x_j)(1 - \sigma(x_j))
\end{aligned}$$

(d) Let $y = f(x)$. From (c) we have $J_y(x) = diag(\sigma'(x))$. Therefore,

$$\begin{aligned} g_x &= diag(\sigma'(x))g_y \\ &= \sigma'(x) \odot g_y \end{aligned}$$

5 Softmax activation function

(a) We show the softmax is invariant under translation. Notice that:

$$\begin{aligned} S(x+c)_i &= \frac{\exp(x_i + c)}{\sum_k \exp(x_k + c)} \\ &= \frac{\exp(c)\exp(x_i)}{\sum_k \exp(c)\exp(x_k)} \\ &= \frac{\exp(x_i)}{\sum_k \exp(x_k)} \\ &= S(x)_i \end{aligned}$$

(b) If we multiply by a scalar greater than 1, we increase the probability of the most probable classes. Likewise, if we multiply by a scalar less than 1, we add probability to the less probable classes.

(c) We show that the two class softmax function can be written as the sigmoid function. Let $z = x_2 - x_1$, then:

$$\begin{aligned} \frac{1}{1+\exp(z)} &= \frac{1}{1+\exp(x_2)\exp(-x_1)} \\ &= \frac{\exp(x_1)}{\exp(x_1)+\exp(x_2)} \end{aligned}$$

And

$$\begin{aligned} 1 - \frac{1}{1+\exp(z)} &= 1 - \frac{\exp(x_1)}{\exp(x_1)+\exp(x_2)} \\ &= \frac{\exp(x_2)}{\exp(x_1)+\exp(x_2)} \end{aligned}$$

(d) We express a k-class softmax as a function f of $k-1$ variables. Define $z_i = x_i - x_1$ $\forall i \neq 1$. Then:

$$f(z_2, \cdots, z_k)_i = \begin{cases} \frac{\exp(z_i)}{\sum_{j=2}^{k} \exp(z_j)+1} & i \neq 1 \\ 1 - \sum_{j=2}^{k} f(z_2, \cdots, z_k)_j & i = 1 \end{cases}$$

We show that $f(z_2, \cdots, z_k)_i = S(x_1, \cdots, c_k)_i$. For $i \neq 1$ we have that:

$$\begin{aligned} f(z_2, \cdots, z_k)_i &= \frac{\exp(x_i - x_1)}{\sum_{j=2}^k \exp(x_j - x_1) + 1} \\ &= \frac{\exp(x_i)}{\sum_{j=2}^k \exp(x_j) + \exp(x_1)} \\ &= S(x_1, \cdots, c_k)_i \end{aligned}$$

For $i = 1$ the result follows since

$$S(x_1, \cdots, x_k)_1 = 1 - \sum_{j=2}^{k} S(x_1, \cdots, x_k)_j = 1 - \sum_{j=2}^{k} f(z_2, \cdots, z_k)_j$$

6 Using cross-entropy cost for real-valued data

(a) We derive the cross entropy cost function using the maximum likelihood principle. Let $X \in \{0, 1\}$ and let $p = P(X = 1)$

$$\begin{aligned} \log \mathcal{L}(p|x) &= \log p^x (1-p)^{1-x} \\ &= x \log p + (1-x) \log(1-p) \end{aligned}$$

(b) A probablistic interpreation of the cross entropy cost function would be the following. Denote the empirical density of X as $\hat{p}(x) = x$ (that is, if $x = 1$ then $p(X = 0) = 1$ and if $x = 0$ then $p(X = 1) = 0$). Let $p(X = 1) = p$ be our model distribution. Then, notice that:

$$\begin{aligned} KL(\hat{p}, p) &= \sum_x \hat{p}(x) \log \frac{\hat{p}(x)}{p(x)} \\ &= \sum_x \hat{p}(x) \log \hat{p}(x) - \sum_x \hat{p}(x) \log p(x) \\ &= -\sum_x \hat{p}(x) \log p(x) \end{aligned}$$

Since $\hat{p}(x) \log \hat{p}(x) = 0$ since if $\hat{p}(x) = 1$, $\log \hat{p}(x) = 0$; and if $\hat{p}(x) = 0$, $0 \log 0 = 0$. We then have:

$$-\sum_x \hat{p}(x) \log p(x) = -x \log p - (1-x) \log(1-p)$$

as needed.

7 Deriving the Glorot initialization scheme

We derive the Glorot initialization scheme from [2] for a ReLU layer using the approach from [3] . As defined in the notes, let:

$$
\begin{aligned}
a^{(k)}(x)_i &= b_i^{(k)} + \sum_{j=1}^{n} w_{ij}^{(k)} h^{(k-1)}(x)_j \\
h^{(k)}(x)_j &= g(a^{(k)}(x)_j) \\
&= \max\left\{0, a^{(k)}(x)_j\right\}
\end{aligned}
$$

We suppose that each $w_{ij}^{(k)}$ is sampled iid from a zero mean symmetric distribution, and that each $h^{(k-1)}(x)_j$ is sampled iid from an independent symmetric distribution. We have that:

$$
\begin{aligned}
Var\left\{a^{(k)}(x)_i\right\} &= Var\left\{\sum_{j=1}^{n_k} w_{ij}^{(k)} h^{(k-1)}(x)_j\right\} \\
&= n_k Var\left\{w^{(k)} h^{(k-1)}\right\} \\
&= n_k \mathbb{E}\left\{(w^{(k)})^2\right\} \mathbb{E}\left\{(h^{(k-1)})^2\right\} - n \underbrace{\mathbb{E}\left\{w^{(k)}\right\}^2}_{=0} \mathbb{E}\left\{h^{(k-1)}\right\}^2 \\
&= n_k Var\left\{w^{(k)}\right\} \mathbb{E}\left\{(h^{(k-1)})^2\right\}
\end{aligned}
$$

Note that we drop the subscripts from both w and h and the argument from h for clarity. If $w^{(k-1)}$ has a symmetric distribution around 0 and that each $b^{(k-1)} = 0$ then $a^{(k-1)}$ is symmetric about 0 and has 0 mean. We have that:

$$
\begin{aligned}
\mathbb{E}\left\{(h^{(k-1)})^2\right\} &= \int_{-\infty}^{\infty} \max\left\{0, a^{(k-1)}\right\}^2 p(a^{(k-1)}) da^{(k-1)} \\
&= \int_{0}^{\infty} (a^{(k-1)})^2 p(a^{(k-1)}) da^{(k-1)} \\
&= \frac{1}{2}\mathbb{E}\left\{(a^{(k-1)})^2\right\} \\
&= \frac{1}{2} Var\left\{a^{(k-1)}\right\}
\end{aligned}
$$

Substituting this into the previous expression yields:

$$
Var\left\{a^{(k)}(x)_i\right\} = \frac{n_k}{2} Var\left\{w^{(k)}\right\} Var\left\{a^{(k-1)}(x)_i\right\}
$$

With L layers we have that:

$$
Var\left\{a^{(L)}(x)_i\right\} = Var\left\{a^{(1)}(x)_i\right\} \prod_{k=1}^{L} \frac{n_k}{2} Var\left\{w^{(k)}\right\}
$$

We see that in order to keep the variances consistent between levels it is enough to have $\frac{n_k}{2}Var\left\{w^{(k)}\right\} = 1\ \forall k$. One way to satisfy this along with the assumptions above is to initialize each $w^{(k)}$ from a $\mathcal{N}\left(0, \sqrt{\frac{2}{n_k}}\right)$ and initialize each $b^{(k)} = 0$.
For the backwards case, we use that:

$$\begin{aligned}\frac{\partial Cost}{\partial h^{(k-1)}(x)_j} &= \sum_{i=1}^{m} w_{ij}^{(k)} \frac{\partial Cost}{\partial a^{(k)}(x)_i} \\ \frac{\partial Cost}{\partial a^{(k)}(x)_i} &= g'(a^{(k)}(x))_i \frac{\partial Cost}{\partial h^{(k)}(x)_i}\end{aligned}$$

We assume that each $w_{ij}^{(k)}$ and $\frac{\partial Cost}{\partial a^{(k)}(x)_i}$ are mutually independent, and each $w_{ij}^{(k)}$ comes from a distribution symmetric around 0 with 0 mean. This implies that $\frac{\partial Cost}{\partial h^{(k-1)}(x)}$ has 0 mean since:

$$\begin{aligned}\mathbb{E}\left\{\frac{\partial Cost}{\partial h^{(k-1)}(x)_i}\right\} &= \mathbb{E}\left\{\sum_{i=1}^{m} w_{ij}^{(k)} \frac{\partial Cost}{\partial a^{(k)}(x)_i}\right\} \\ &= \sum_{i=1}^{m} \underbrace{\mathbb{E}\left\{w_{ij}^{(k)}\right\}}_{=0} \mathbb{E}\left\{\frac{\partial Cost}{\partial a^{(k)}(x)}\right\} \\ &= 0\end{aligned}$$

We assume that $g'(a^{(k)})$ and $\frac{\partial Cost}{\partial h^{(k)}}$ are independent (dropping the subscripts and arguments once more) to get:

$$\begin{aligned}\mathbb{E}\left\{\frac{\partial Cost}{\partial a^{(k)}}\right\} &= \mathbb{E}\left\{g'(a^{(k)})\right\} \mathbb{E}\left\{\frac{\partial Cost}{\partial h^{(k)}}\right\} \\ &\overset{(a)}{=} \frac{1}{2}\mathbb{E}\left\{\frac{\partial Cost}{\partial h^{(k)}}\right\} \\ &\overset{(b)}{=} 0\end{aligned}$$

Where (a) follows since $g'(a^{(k)}) = H(a^{(k)}) = 1$ wp $\frac{1}{2}$ and is 0 otherwise, and (b) follows from the previous claim. We then get:

$$\begin{aligned}Var\left\{\frac{\partial Cost}{\partial a^{(k)}}\right\} &= \mathbb{E}\left\{\left(\frac{\partial Cost}{\partial a^{(k)}}\right)^2\right\} \\ &= \int_{-\infty}^{\infty} \left(H(a^{(k)})\frac{\partial Cost}{\partial h^{(k)}}\right)^2 p(a^{(k)})da^{(k)} \\ &= \int_{0}^{\infty} \frac{\partial Cost}{\partial h^{(k)}}^2 p(a^{(k)})da^{(k)} \\ &= \frac{1}{2}Var\left\{\frac{\partial Cost}{\partial h^{(k)}}\right\}\end{aligned}$$

We compute the variance as before:

$$Var\left\{\frac{\partial Cost}{\partial h^{(k-1)}}\right\} = \frac{m_k}{2} Var\left\{w^{(k)}\right\} Var\left\{\frac{\partial Cost}{\partial h^{(k)}}\right\}$$

With L layers we have that:

$$Var\left\{\frac{\partial Cost}{\partial h^{(2)}}\right\} = Var\left\{\frac{\partial Cost}{\partial h^{(L+1)}}\right\} \prod_{k=1}^{L} \frac{m_k}{2} Var\left\{w^{(k)}\right\}$$

Using identical reasoning as with the forward case, it is enough to initialize each $w^{(k)}$ from a $\mathcal{N}\left(0, \sqrt{\frac{2}{m_k}}\right)$ and initialize each $b^{(k)} = 0$.

References

[1] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[2] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics* (Y. W. Teh and M. Titterington, eds.), vol. 9 of *Proceedings of Machine Learning Research*, (Chia Laguna Resort, Sardinia, Italy), pp. 249–256, PMLR, 13–15 May 2010.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, (Washington, DC, USA), pp. 1026–1034, IEEE Computer Society, 2015.