ESL: 11.1

From The Neural network side, since there is ~~not~~ one hidden layer.

$$f_k(X) = \beta_0 + \beta_k^T \theta(\alpha_{0m} + \alpha_m^T X)$$

From The PPR side we know:

$$f(x) = \sum_{m=1}^{N} g_m(w_m^T X)$$

Also we know:

$$g_m(w_m^T X) = \beta_m \theta(\alpha_{0m} + S_m(w_m^T X))$$

Then since is a PPR we know that $w = \frac{\hat{w}}{\|\hat{w}\|}$,

Then, we need to find some S_m and $\hat{w}$, such will allow us to equiparate Both ~~modules~~; this is $S = \|\alpha_m\|$ a

$$w_m^T = \frac{\alpha_m}{\|\alpha_m\|}$$

Then $g_m(w^T x) = f_m(x) = \beta_0 + \beta_m \theta(\alpha_{0m} + \alpha_m^T x)$ and

$$f(x) = \sum_{k=1}^{K} f_m(x).$$

The same argument could be done if we use a logistic regression for the $g_m(w_m^T x)$ of PPR and a classification network from the neural network where $g(T) = \frac{e^{T_k}}{\sum_{l=1} e^{T_l}}$.

ESL: 11.2

If the function is linear we know that $f''=0$.

Then if $y = \frac{1}{1+e^{-v}}$

$$\frac{\partial y}{\partial v} = \frac{e^{-v}}{(1+e^{-v})^2}$$

$$\frac{\partial^2 y}{\partial v^2} = e^{-v}(1+e^{-v})^2\left(2\cdot(1+e^{-v})^{-1} - 1\right)$$

Then if $v=0$.

$$\left.\frac{\partial^2 y}{\partial v^2}\right|_{v=0} = \frac{1}{4}\cdot\left(\frac{2}{2} - 1\right)$$

$= 0$ // $\Rightarrow$ The we could argue that is lined at $v=0$.

ESL 11.4

$$R(\theta) = -\sum_{i=1}^{N} \sum_{k=1}^{K} y_{ik} \log f_k(x_i)$$

$$f_k(X) = \frac{e^{T_k}}{\sum_{l=1}^{K} e^{T_l}}$$

Since there is no hidden layer

$$T_k = \beta_{0k} + \beta_k^T X.$$

And now we assume that the $k=1$ is base category. Then

$$T_k - T_1 = (\beta_{0k} - \beta_{01}) + (\beta_k - \beta_1)^T X$$

$$\therefore \Pr(K = k) = f_k(X) = \frac{e^{T_k - T_1}}{1 + e^{T_k - T_1}}$$

Wich transform for multiple K and using cross entropy.

$$-\sum y_{ik} \log(f_k(X))$$

$\Rightarrow$ wich is the multilogistic objective function.

5.1 Bishop

First we need to found the conversion or relation between $\tanh(a)$ and $\sigma(a)$

$$\frac{e^a - e^{-a}}{e^a + e^{-a}} = \frac{e^a}{e^a + e^{-a}} - \frac{e^{-a}}{e^a + e^{-a}} = \frac{e^a}{e^a + e^{-a}} - \frac{e^{-a} + e^a - e^a}{e^a + e^{-a}}$$

$$= \frac{e^a}{e^a + e^{-a}} - \left(1 - \frac{e^a}{e^a + e^{-a}}\right)$$

$$= \frac{2e^a}{e^a + e^{-a}} \cdot \frac{e^{-a}}{e^{-a}} - 1$$

$$= \frac{2}{1 + e^{-2a}} - 1$$

$$\frac{e^a - e^{-a}}{e^a + e^{-a}} = 2\,\sigma(2a) - 1$$

Then if we replace this into

$$a_k = \sum_{j=1}^{n} w_{kj}^{(k)} \cdot (2\sigma(2a) - 1) + w_{k0}^{(k)}$$

$$a_k = \sum_{j=1}^{n} 2w^{(k)} \sigma(2a) - 2w_{kj}^{k} + w_{k0}^{(k)}$$

$$a_k = \sum_{j}^{n} 2w_{kj}^{(k)} \sigma(2a) + \sum_{j}^{n} \left[-2w_{kj}^{k} + w_{k0}^{k} \right]$$

Then if for a regular neural network ~~[illegible]~~ we have to convert into:

$$w_p^{(2p)} = 2w_k^{(2)}, \quad a_j^{(p)} = 2a_j^{(2)}, \quad w_{k0}^{(2p)} = -\sum_{j=1}^{n} w_{kj}^{(2)} + w_{k0}^{2}$$

Using this equality we transform a_k into:

$$\boxed{a_p = \sum_{j=1}^{n} w_p^{(2p)} \theta(a) + w_{p0}^{(2p)}}$$

Bishop 5.5

$$P(T/W_1,\dots,W_K) = \prod_{K=1}^{K} y_K^{t_K} \Rightarrow$$ Network output for K class.

Then for N data point we have that.

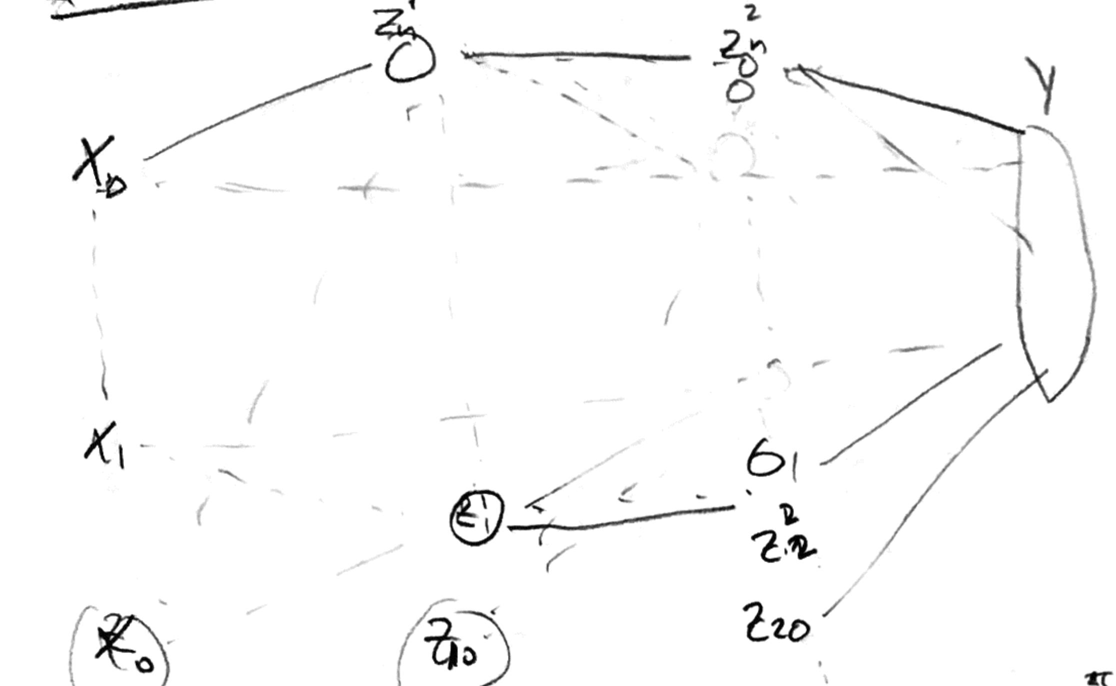
$$p(T/W_1,\dots,W_K) = \prod_{K=1}^{K}\prod_{n=1}^{N} y_{nK}^{t_{nK}}$$

Then if we compute for the log function.

$$l(T/W_1,\dots,W_K) = -\sum_{K=1}^{K}\sum_{n=1}^{N} t_{nK}\cdot \log(y_{nK}).$$

Which is the log likelihood function of the 5.24.

Question 6



$$Z_n^2 = \sigma\left(\beta_{0k}^{(2)} + \beta^{T(2)} Z_n^1\right)$$

Equations:

$$Z_n^1 = \sigma(\underbrace{\alpha_{0n} + \alpha_n^T X}_{P})$$

$$\hat{Y}_k = \sigma\left(\beta_{0k}^1 + \beta_k^{(1)T} Z^2 + \gamma_k^T X + \gamma_{0k}\right)$$

$$R = \frac{1}{n}\sum_{i=1}^{K} \sigma\left(\hat{Y}_k - y\right)^2$$

$$\frac{\partial R}{\partial \beta_k} = \delta_{ki} Z_{ni}, \quad s_{mi} = \sigma'(\alpha_n^T x_i) \sum_{k=1}^{K} \beta_{kn} \delta_{ki}$$

δ_{ki} => are errors in output layer

$$\frac{\partial R}{\partial \alpha_n} = s_{ni} x_{il}$$

$$\frac{\partial R}{\partial \gamma_k} = 2(\hat{y}_n - y) \cdot \hat{y}_k (1 - \hat{y}_k) \cdot X$$

* The main change is that we have a another set of parameters that go from the input layer to the output layer, creating a new partial derivative.