Mid-term exam solution key

Prayag
Neural networks and learning systems-I

May 21, 2019

Problem 1.

Solution. 1. Consider the update equation

$$\overline{W}(n+1) = \overline{W}(n) - \eta\Delta\overline{W} \tag{1}$$

where $\overline{W}(n)$ is the weight vector at time n, η is the learning rate, and $\Delta\overline{W}$ is the gradient of the cost function with respect to the weight vector $\overline{W}(n)$.

(a) The learning rate must be increased when the derivative of the cost function with respect to a weight vector has the same algebraic sign to accelerate the convergence of the algorithm.

(b) The learning rate must be decreased when the algebraic sign of the derivative of cost function alternates with consecutive iterations. This reduces the oscillations during the convergence fo the algorithm.

2. This follows from the composition of linear maps. Consider a affine linear map $g(x) = ax + b$ where a and b are non-zero constants. Similarly $f(x) = a'x + b'$ with a' and b' being non-zero constants. Consider the composition $f(g(x)) = cx + d$ where $c = aa'$ and $d = a'b + d$ which is again a affine linear map.

3. Linear regression: $Y = \sigma_0 + \sigma_1 X + \epsilon$, Quartic regression: $Y = \sigma_0 + \sigma_1 X + \sigma_2 X^2 + \sigma_3 X^3 + \sigma_4 X^4 \epsilon$, LRSS: training residual sum of squares for linear regression, and QRSS: training residual sum of squares for quartic regression.
Case 1: Let $\epsilon \neq 0$ and $\mathbb{E} = 0$. In the case of linear regression, $Y = f(x) + \epsilon$ we have

$$\begin{aligned}
\mathbb{E}\left((y-\hat{y})^2\right) =& \mathbb{E}\left(f(x) + \epsilon - \hat{f}(x)\right)^2 \\
=& \mathbb{E}\left(f(x) - \hat{f}(x)\right)^2 + \mathbb{E}(\epsilon^2) + 2\mathbb{E}(\epsilon)\mathbb{E}\left(f(x) - \hat{f}(x)\right) \\
=& \underbrace{\mathbb{E}\left(f(x) - \hat{f}(x)\right)^2}_{\text{can be minimized}} + \underbrace{\mathrm{Var}(x)}_{\text{cannot be minimized}}
\end{aligned} \tag{2}$$

Since $\mathrm{Var}(x)$ cannot be minimized using linear regression and on the other hand the quartic regression is more flexible and can even fit the points with noise, therefore

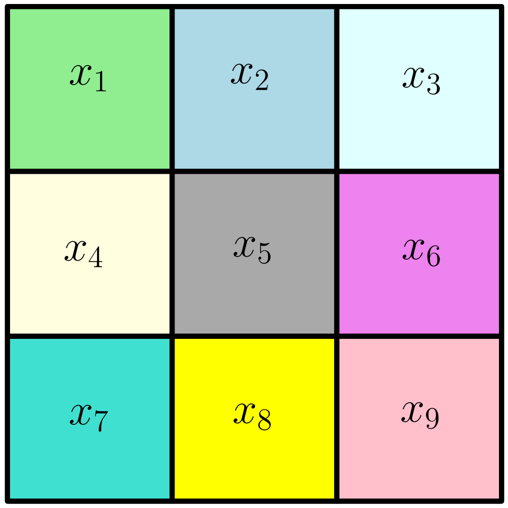

Figure 1: A 3×3 image.

QRSS $\leq$ LRSS.
Case 2: With $\epsilon = 0$, the relationship is truly linear i.e., $Y = \sigma_0 + \sigma_1 X$. In such cases, LRSS and QRSS will be equal since both of the models can fit the data exactly.

4. Consider a 3×3 image as shown in Figure 1 and a 2×2 kernel as shown in Figure 2. Moving the kernel over the 3×3 image we get 4 outputs which is given by

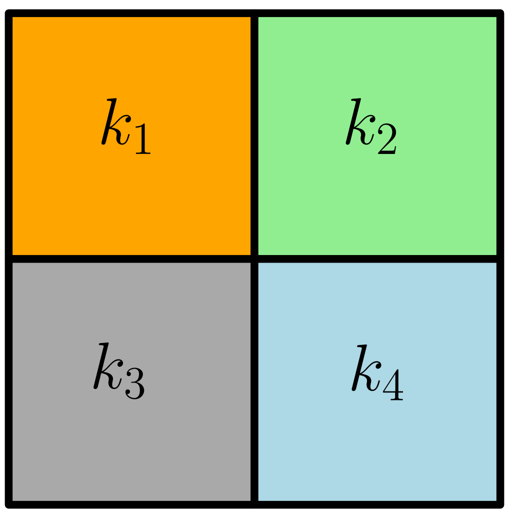

Figure 2: A 2×2 kernel.

$$
\begin{aligned}
a_1 =& x_1 k_1 + x_2 k_2 + x_4 k_3 + x_5 k_4 \\
a_2 =& x_2 k_1 + x_3 k_2 + x_5 k_3 + x_6 k_4 \\
a_3 =& x_4 k_1 + x_5 k_2 + x_7 k_3 + x_8 k_4 \\
a_4 =& x_5 k_1 + x_6 k_2 + x_8 k_3 + x_9 k_4.
\end{aligned}
$$

Graphical illustration of all the connections is shown in Figure 3

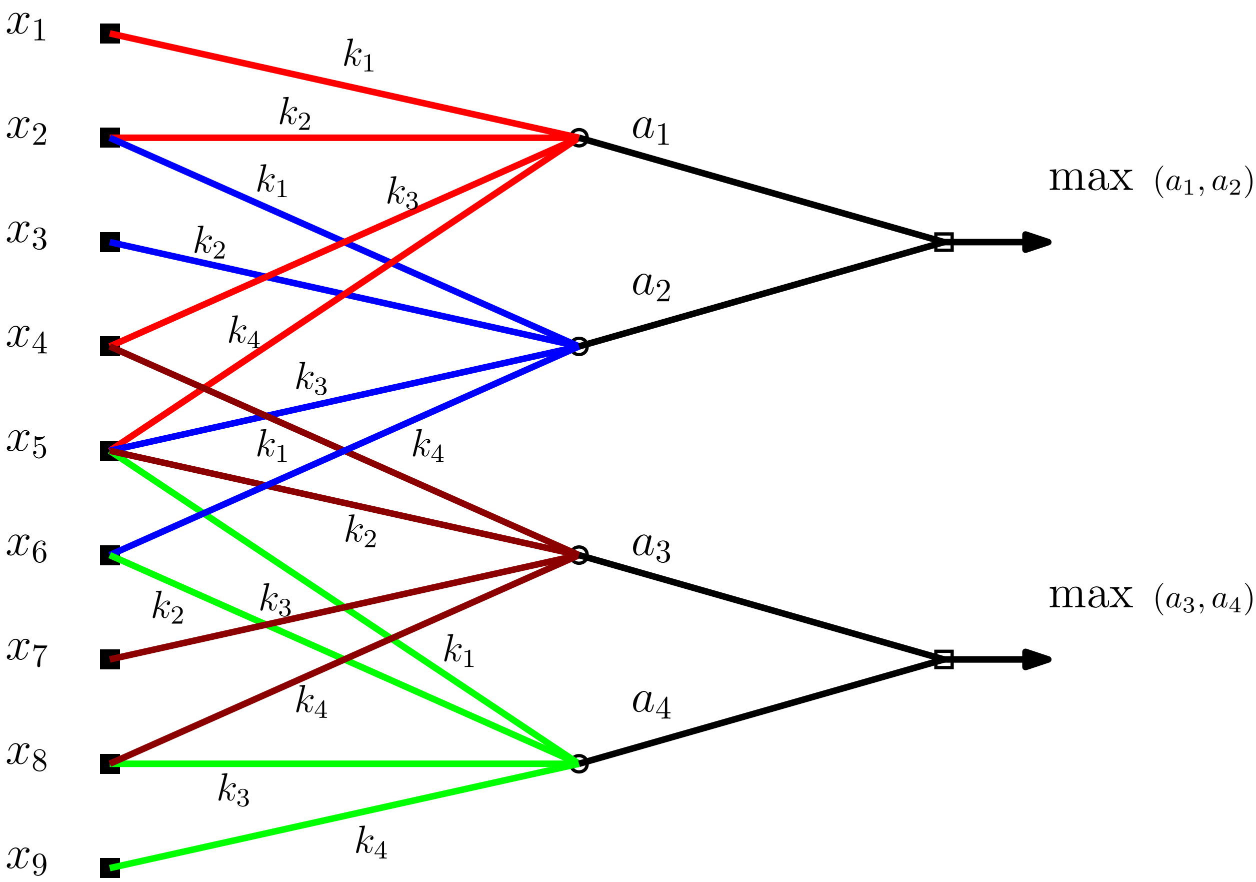

Figure 3

$\blacksquare$

Problem 2.

Solution. The intermediate variables in the network shown in Figure 4 is given by

$$\begin{aligned}
v_1 =& x_1 + x_2 + x_3 - 0.5 \\
v_2 =& x_1 + x_2 + x_3 - 1.5 \\
v_3 =& x_1 + x_2 + x_3 - 2.5 \\
y_1 =& \phi(v_1) \\
y_2 =& \phi(v_2) \\
y_3 =& \phi(v_3)
\end{aligned} \tag{3}$$

where

$$\begin{cases} 1 & x \geq 0 \\ 0 & \text{Otherwise} \end{cases} \tag{4}$$

$\blacksquare$

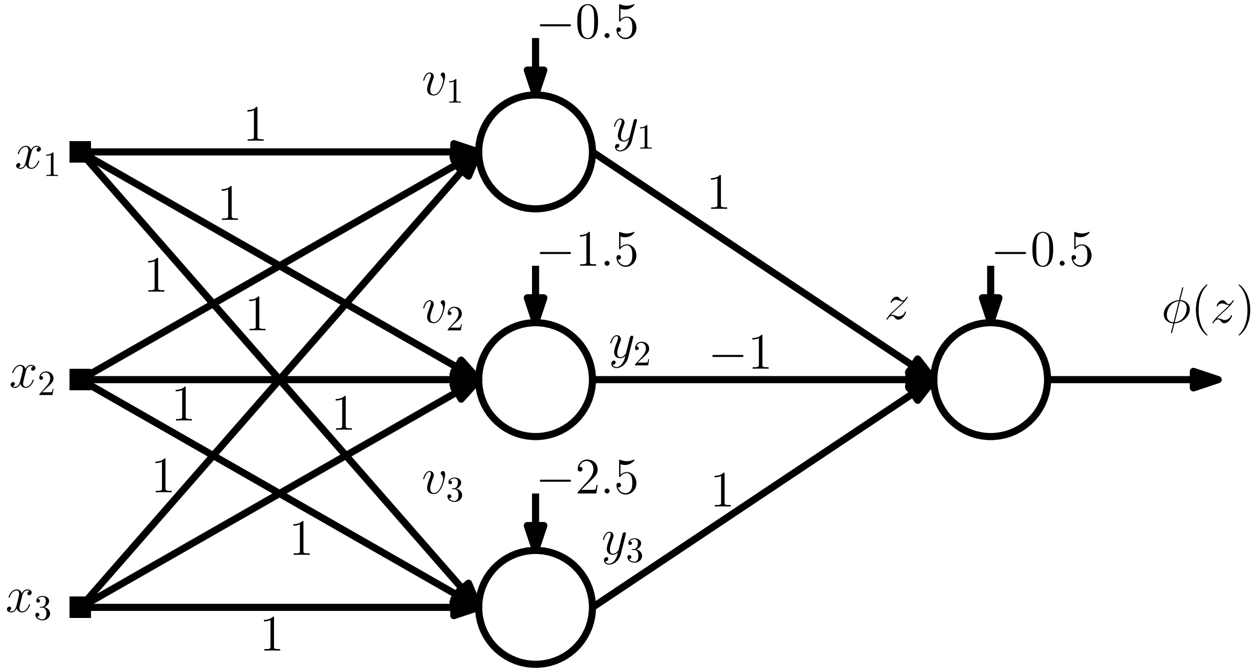

Figure 4: 3-bit XOR network.

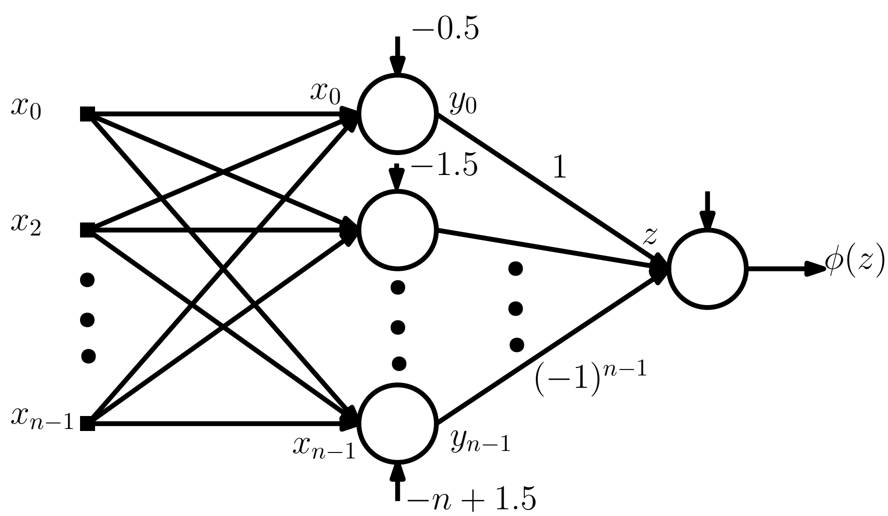

Figure 5: n-bit XOR network.

Table 1: Intermediate variables in the network

x_1	x_2	x_3	v_1	v_2	v_3	y_1	y_2	y_3	z	$\phi(z)$
0	0	0	-0.5	-1.5	-2.5	0	0	0	-0.5	0
0	0	1	0.5	-0.5	-1.5	1	0	0	0.5	1
0	1	0	0.5	-0.5	-1.5	1	0	0	0.5	1
0	1	1	1.5	0.5	-0.5	1	1	0	-0.5	0
1	0	0	0.5	-1.5	-1.5	1	0	0	0.5	1
1	0	1	1.5	0.5	-0.5	1	1	0	-0.5	0
1	1	0	1.5	0.5	-0.5	1	1	0	-0.5	0
1	1	1	2.5	1.5	0.5	1	1	1	0.5	1

Problem 3.

Solution. The cost function is given by

$$J(\overline{W}) = \sum_{x \in \mathscr{H}} \left(-\overline{W}^{\mathrm{T}} x \right) \tag{5}$$

where $\mathscr{H}$ is the of misclassified inputs. Differentiating $J(\overline{W})$ with respect to $\overline{W}(n)$ we get

$$\nabla_{\overline{W}} J(\overline{W}) = \sum_{x \in \mathscr{H}} -x \tag{6}$$

The weight vector is updated as follows:

$$\begin{aligned} \overline{W}(n+1) =& \overline{W}(n) - \eta(n) \nabla_{\overline{W}} J(\overline{W}) \\ =& \overline{W}(n) + \eta \sum_{x \in \mathscr{H}} x \end{aligned} \tag{7}$$

where η is assumed to remain same for all n, say $\eta = 1$. Let the initial weight vector be $\overline{W}(0) = \overline{0}$. Consider $\overline{W}^{\mathrm{T}} \overline{x} \leq 0$ and $\overline{x}(n) \in \mathscr{H}$ is the set of misclassified samples. We know that

$$\begin{aligned} \overline{W}(n+1) =& \overline{W}(n) + \sum_{\overline{x} \in \mathscr{H}} \overline{x} \\ =& \sum_{\overline{x} \in \mathscr{H}_0} \overline{x} + \cdots + \sum_{\overline{x} \in \mathscr{H}_n} \overline{x} \end{aligned} \tag{8}$$

let us consider a $\overline{W}_0$ such that $\overline{W}_0^{\mathrm{T}} \overline{x} > 0$ for all $\overline{x}(n)$ belongs to class 1. Pre-multiplying the above equation by $\overline{W}_0^{\mathrm{T}}$ we get

$$\overline{W}_0^{\mathrm{T}} \overline{W}(n+1) = \sum_{\overline{x} \in \mathscr{H}_0} \overline{W}_0^{\mathrm{T}} \overline{x} + \cdots + \sum_{\overline{x} \in \mathscr{H}_n} \overline{W}_0^{\mathrm{T}} \overline{x}. \tag{9}$$

Let $\alpha = \min_i \sum_{\overline{x} \in \mathscr{H}_i} \overline{W}_0^{\mathrm{T}} \overline{x}$. Using Cauchy-Schwartz inequality we get

$$
\begin{aligned}
\|\overline{W}_0\|^2 \|\overline{W}_{n+1}\|^2 &\geq (n+1)^2 \alpha^2 \\
\|\overline{W}_{n+1}\|^2 &\geq \frac{(n+1)^2 \alpha^2}{\|\overline{W}_0\|^2}
\end{aligned} \tag{10}
$$

We know that,

$$
\begin{aligned}
\overline{W}(n+1) =& \overline{W}(n) + \sum_{\overline{x} \in \mathscr{H}_n} \overline{x} \\
\|\overline{W}(n+1)\|^2 =& \|\overline{W}(n)\|^2 + \| \sum_{\overline{x} \in \mathscr{H}_n} \overline{x}\|^2 + 2 \underbrace{\overline{W}^{\mathrm{T}}(n) \sum_{\overline{x} \in \mathscr{H}_n} \overline{x}}_{\overline{W}^{\mathrm{T}}(n)\overline{x} \leq 0} \\
\|\overline{W}(n+1)\|^2 \leq& \|\overline{W}(n)\|^2 + \| \sum_{\overline{x} \in \mathscr{H}_n} \overline{x}\|^2 \\
\leq& \sum_{1}^{n} \| \sum_{\overline{x} \in \mathscr{H}_n} \overline{x}\|^2 \\
\leq& (n+1)\beta
\end{aligned} \tag{11}
$$

where $\beta = \max_i \| \sum_{\overline{x} \in \mathscr{H}_i} \overline{x}\|^2$. Using equations (10) and (11) we get

$$
n_{\max} = \left(\frac{\beta}{\alpha^2} \|\overline{W}_0\|^2 \right) - 1. \tag{12}
$$

Therefore the batch perceptron algorithm converges after $n_{\max}$ epochs. ■

Problem 4.

Solution. Consider through the Taylor series expansion

$$
E_{\mathrm{avg}}\left(\overline{W}(n) + \Delta\overline{W}(n)\right) = E_{\mathrm{avg}}\left(\overline{W}(n)\right) + \overline{g}^{\mathrm{T}}(n)\Delta\overline{W}(n) + \frac{1}{2}\Delta\overline{W}(n)^{\mathrm{T}}\mathbf{H}(n)\Delta\overline{W}(n) + \text{h.o.t} \tag{13}
$$

and neglecting the h.o.t we get

$$
E_{\mathrm{avg}}\left(\overline{W}(n) + \Delta\overline{W}(n)\right) = E_{\mathrm{avg}}\left(\overline{W}(n)\right) + \overline{g}^{\mathrm{T}}(n)\Delta\overline{W}(n) + \frac{1}{2}\Delta\overline{W}(n)^{\mathrm{T}}\mathbf{H}(n)\Delta\overline{W}(n). \tag{14}
$$

Differentiating the above equation with respect to $\Delta\overline{W}(n)^{\mathrm{T}}$ we get

$$
\begin{aligned}
\frac{\partial E_{\mathrm{avg}}\left(\overline{W}(n) + \Delta\overline{W}(n)\right)}{\partial \Delta\overline{W}(n)^{\mathrm{T}}} =& 0 \\
\mathbf{H}(n)\Delta\overline{W}(n) =& -\overline{g}(n) \\
\Delta\overline{W}(n)^{\mathrm{T}} =& -\mathbf{H}^{-1}(n)\overline{g}(n)
\end{aligned} \tag{15}
$$

provided $\mathbf{H}^{-1}(n)$ exists. One can also get a pseudo-inverse of $\mathbf{H}$ in case of singularity. The advantages of Hessian are as follows:

1. Accelerated convergence.
2. Possibly low rank approximations over $\mathbf{H}$ to obtain low complexity algorithms (i.e., there is control on complexity).

$\blacksquare$