شبکه‌های عصبی مصنوعی

جلسه نهم:
پرسپترون چند لایه (۵)
(Multi-Layer Perceptron = MLP)

محمد فرخی (farrokhi AT iust.ac.ir)

پرسپترون چندلایه (MLP)

پرسپترون چندلایه (MLP)

یافتن ساختار بهینه برای MLP:

پرسپترون چندلایه (MLP)

یافتن ساختار بهینه برای MLP:

- به دو روش می‌توان این‌کار را انجام داد:

پرسپترون چندلایه (MLP)

یافتن ساختار بهینه برای MLP:

- به دو روش می‌توان این‌کار را انجام داد:

۱- رشددادن شبکه (Network Growing):

با تعداد کمی سلول در لایه(های) پنهان شروع‌کرده و به‌تدریج سلول یا لایه پنهان به آن اضافه می‌کنیم.

پرسپترون چندلایه (MLP)

یافتن ساختار بهینه برای MLP:

- به دو روش می‌توان این‌کار را انجام داد:

۱- رشددادن شبکه (Network Growing):

با تعداد کمی سلول در لایه(های) پنهان شروع‌کرده و به‌تدریج سلول یا لایه پنهان به آن اضافه می‌کنیم.

۲- هرس‌کردن شبکه (Network Pruning):

با یک شبکه بزرگ شروع کرده و سپس وزن‌های معینی را تضعیف یا قطع می‌کنیم.

پرسپترون چندلایه (MLP)

یافتن ساختار بهینه برای MLP:

- به دو روش می‌توان این‌کار را انجام داد:

۱- رشددادن شبکه (Network Growing):

با تعداد کمی سلول در لایه(های) پنهان شروع‌کرده و به‌تدریج سلول یا لایه پنهان به آن اضافه می‌کنیم.

۲- هرس‌کردن شبکه (Network Pruning):

با یک شبکه بزرگ شروع کرده و سپس وزن‌های معینی را تضعیف یا قطع می‌کنیم.

استفاده از ماتریس هِس (Hessian Matrix) **سطح خطا برای هرس‌کردن شبکه:**

پرسپترون چندلایه (MLP)

یافتن ساختار بهینه برای MLP:

– به دو روش می‌توان این‌کار را انجام داد:

۱– رشددادن شبکه (Network Growing):

با تعداد کمی سلول در لایه(های) پنهان شروع‌کرده و به‌تدریج سلول یا لایه پنهان به آن اضافه می‌کنیم.

۲– هرس‌کردن شبکه (Network Pruning):

با یک شبکه بزرگ شروع کرده و سپس وزن‌های معینی را تضعیف یا قطع می‌کنیم.

استفاده از ماتریس هِس (Hessian Matrix) سطح خطا برای هرس‌کردن شبکه:

- سطح خطا، تابعی است از وزن‌های شبکه $E(\mathbf{w})$

پرسپترون چندلایه (MLP)

یافتن ساختار بهینه برای MLP:

– به دو روش می‌توان این‌کار را انجام داد:

۱– رشددادن شبکه (Network Growing):

با تعداد کمی سلول در لایه(های) پنهان شروع‌کرده و به‌تدریج سلول یا لایه پنهان به آن اضافه می‌کنیم.

۲– هرس‌کردن شبکه (Network Pruning):

با یک شبکه بزرگ شروع کرده و سپس وزن‌های معینی را تضعیف یا قطع می‌کنیم.

استفاده از ماتریس هِس (Hessian Matrix) سطح خطا برای هرس‌کردن شبکه:

- سطح خطا، تابعی است از وزن‌های شبکه $E(\mathbf{w})$
- استفاده از گرادیان دوم سطح خطا نسبت به وزن‌ها برای حذف وزن‌های «کمترموثر»

پرسپترون چندلایه (MLP)

یافتن ساختار بهینه برای MLP:

– به دو روش می‌توان این‌کار را انجام داد:

۱– رشددادن شبکه (Network Growing):

با تعداد کمی سلول در لایه(های) پنهان شروع‌کرده و به‌تدریج سلول یا لایه پنهان به آن اضافه می‌کنیم.

۲– هرس‌کردن شبکه (Network Pruning):

با یک شبکه بزرگ شروع کرده و سپس وزن‌های معینی را تضعیف یا قطع می‌کنیم.

استفاده از ماتریس هِس (Hessian Matrix) سطح خطا برای هرس‌کردن شبکه:

- سطح خطا، تابعی است از وزن‌های شبکه $E(\mathrm{w})$
- استفاده از گرادیان دوم سطح خطا نسبت به وزن‌ها برای حذف وزن‌های «کمترموثر»
- استفاده از روش اختلال (Perturbation Method) برای یافتن وزن‌های «کمتر موثر»

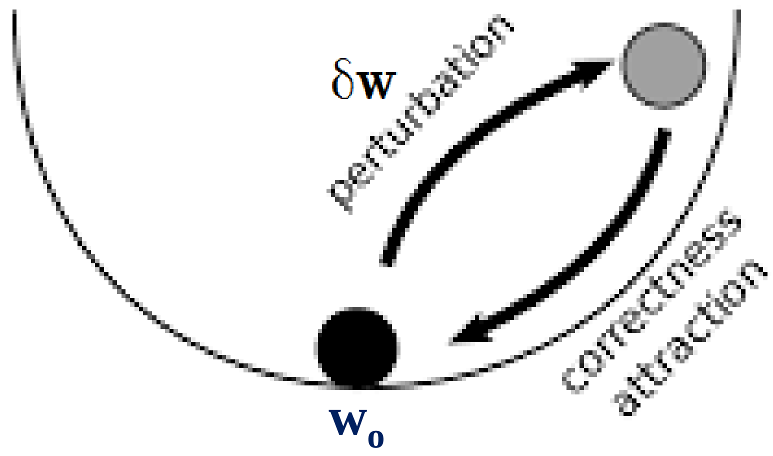
پرسپترون چندلایه (MLP)

استفاده از ماتریس هِس (Hessian Matrix) سطح خطا برای هرس‌کردن شبکه:

پرسپترون چندلایه (MLP)

استفاده از ماتریس هِس (Hessian Matrix) سطح خطا برای هرس‌کردن شبکه:

– فرض‌کنید وزن‌ها همگرا شده باشند ($\mathbf{w}_o$) و این وزن‌ها را به‌مقدار اندکی ($\delta\mathbf{w}$) تغییر (اختلال) می‌دهیم.



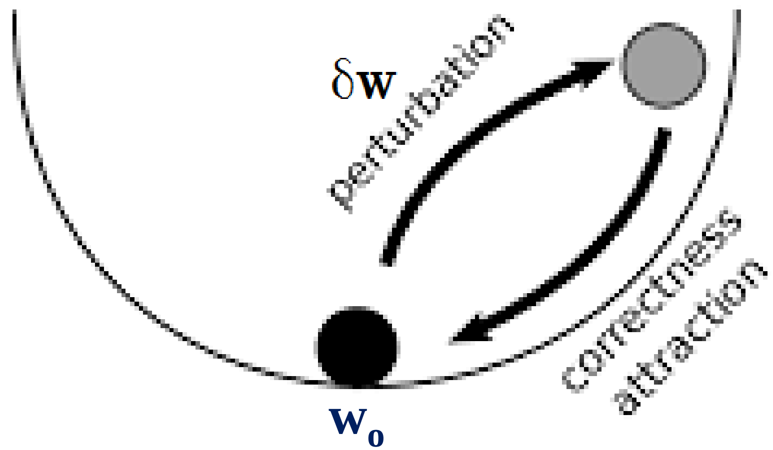
پرسپترون چندلایه (MLP)

استفاده از ماتریس هِس (Hessian Matrix) سطح خطا برای هرس‌کردن شبکه:

– فرض‌کنید وزن‌ها همگرا شده باشند ($\mathbf{w}_o$) و این وزن‌ها را به‌مقدار اندکی ($\delta\mathbf{w}$) تغییر (اختلال) می‌دهیم.

– بسط تیلور تابع غیرخطی خطا حول $\mathbf{w}_o$

$$E(\mathbf{w}_o + \delta\mathbf{w}) = E(\mathbf{w}_o) + (\nabla_{\mathbf{w}} E\big|_{\mathbf{w}=\mathbf{w}_o})^T \delta\mathbf{w} + \frac{1}{2!}\delta\mathbf{w}^T (\nabla^2_{\mathbf{w}} E\big|_{\mathbf{w}=\mathbf{w}_o})\,\delta\mathbf{w} + \mathrm{H.O.T.}$$



پرسپترون چندلایه (MLP)

استفاده از ماتریس هِس (Hessian Matrix) سطح خطا برای هرس‌کردن شبکه:

– فرض‌کنید وزن‌ها همگرا شده باشند ($\mathbf{w_o}$) و این وزن‌ها را به‌مقدار اندکی ($\delta\mathbf{w}$) تغییر (اختلال) می‌دهیم.

– بسط تیلور تابع غیرخطی خطا حول $\mathbf{w_o}$

$$E(\mathbf{w}_o + \delta\mathbf{w}) = E(\mathbf{w}_o) + (\nabla_{\mathbf{w}} E\big|_{\mathbf{w}=\mathbf{w}_o})^T \delta\mathbf{w} + \frac{1}{2!}\delta\mathbf{w}^T (\nabla^2_{\mathbf{w}} E\big|_{\mathbf{w}=\mathbf{w}_o})\,\delta\mathbf{w} + \mathrm{H.O.T.}$$

پرسپترون چندلایه (MLP)

استفاده از ماتریس هِس (Hessian Matrix) سطح خطا برای هرس‌کردن شبکه:

– فرض‌کنید وزن‌ها همگرا شده باشند ($\mathbf{w}_o$) و این وزن‌ها را به‌مقدار اندکی ($\delta\mathbf{w}$) تغییر (اختلال) می‌دهیم.

– بسط تیلور تابع غیرخطی خطا حول $\mathbf{w}_o$

$$E(\mathbf{w}_o + \delta\mathbf{w}) = E(\mathbf{w}_o) + (\nabla_{\mathbf{w}} E\big|_{\mathbf{w}=\mathbf{w}_o})^T \delta\mathbf{w} + \frac{1}{2!}\delta\mathbf{w}^T (\nabla^2_{\mathbf{w}} E\big|_{\mathbf{w}=\mathbf{w}_o})\delta\mathbf{w} + \mathrm{H.O.T.}$$

$$E(\mathbf{w}_o) + \delta E = E(\mathbf{w}_o) + (\nabla_{\mathbf{w}} E\big|_{\mathbf{w}=\mathbf{w}_o})^T \delta\mathbf{w} + \frac{1}{2!}\delta\mathbf{w}^T (\nabla^2_{\mathbf{w}} E\big|_{\mathbf{w}=\mathbf{w}_o})\delta\mathbf{w} + \mathrm{H.O.T.}$$

پرسپترون چندلایه (MLP)

استفاده از ماتریس هِس (Hessian Matrix) سطح خطا برای هرس‌کردن شبکه:

– فرض‌کنید وزن‌ها همگرا شده باشند ($\mathbf{w}_o$) و این وزن‌ها را به‌مقدار اندکی ($\delta\mathbf{w}$) تغییر (اختلال) می‌دهیم.

– بسط تیلور تابع غیرخطی خطا حول $\mathbf{w}_o$

$$E(\mathbf{w}_o + \delta\mathbf{w}) = E(\mathbf{w}_o) + (\nabla_{\mathbf{w}} E\big|_{\mathbf{w}=\mathbf{w}_o})^T \delta\mathbf{w} + \frac{1}{2!}\delta\mathbf{w}^T (\nabla^2_{\mathbf{w}} E\big|_{\mathbf{w}=\mathbf{w}_o})\delta\mathbf{w} + \mathrm{H.O.T.}$$

$$\cancel{E(\mathbf{w}_o)} + \delta E = \cancel{E(\mathbf{w}_o)} + (\nabla_{\mathbf{w}} E\big|_{\mathbf{w}=\mathbf{w}_o})^T \delta\mathbf{w} + \frac{1}{2!}\delta\mathbf{w}^T (\nabla^2_{\mathbf{w}} E\big|_{\mathbf{w}=\mathbf{w}_o})\delta\mathbf{w} + \mathrm{H.O.T.}$$

پرسپترون چندلایه (MLP)

استفاده از ماتریس هِسِ (Hessian Matrix) سطح خطا برای هرس‌کردن شبکه:

– فرض‌کنید وزن‌ها همگرا شده باشند ($\mathbf{w}_o$) و این وزن‌ها را به‌مقدار اندکی ($\delta\mathbf{w}$) تغییر (اختلال) می‌دهیم.

– بسط تیلور تابع غیرخطی خطا حول $\mathbf{w}_o$

$$E(\mathbf{w}_o + \delta\mathbf{w}) = E(\mathbf{w}_o) + (\nabla_{\mathbf{w}} E\big|_{\mathbf{w}=\mathbf{w}_o})^T \delta\mathbf{w} + \frac{1}{2!}\delta\mathbf{w}^T (\nabla^2_{\mathbf{w}} E\big|_{\mathbf{w}=\mathbf{w}_o})\,\delta\mathbf{w} + \mathrm{H.O.T.}$$

$$\cancel{E(\mathbf{w}_o)} + \delta E = \cancel{E(\mathbf{w}_o)} + (\nabla_{\mathbf{w}} E\big|_{\mathbf{w}=\mathbf{w}_o})^T \delta\mathbf{w} + \frac{1}{2!}\delta\mathbf{w}^T (\nabla^2_{\mathbf{w}} E\big|_{\mathbf{w}=\mathbf{w}_o})\,\delta\mathbf{w} + \mathrm{H.O.T.}$$

$$\delta E = (\nabla_{\mathbf{w}} E\big|_{\mathbf{w}=\mathbf{w}_o})^T \delta\mathbf{w} + \frac{1}{2!}\delta\mathbf{w}^T (\nabla^2_{\mathbf{w}} E\big|_{\mathbf{w}=\mathbf{w}_o})\,\delta\mathbf{w} + \mathrm{H.O.T.}$$

پرسپترون چندلایه (MLP)

استفاده از ماتریس هِس (Hessian Matrix) سطح خطا برای هرس‌کردن شبکه:

– فرض‌کنید وزن‌ها همگرا شده باشند ($\mathbf{w}_0$) و این وزن‌ها را به‌مقدار اندکی ($\delta\mathbf{w}$) تغییر (اختلال) می‌دهیم.

– بسط تیلور تابع غیرخطی خطا حول $\mathbf{w}_0$

$$E(\mathbf{w}_o + \delta\mathbf{w}) = E(\mathbf{w}_o) + (\nabla_{\mathbf{w}} E\big|_{\mathbf{w}=\mathbf{w}_o})^T \delta\mathbf{w} + \frac{1}{2!}\delta\mathbf{w}^T (\nabla^2_{\mathbf{w}} E\big|_{\mathbf{w}=\mathbf{w}_o})\delta\mathbf{w} + \mathrm{H.O.T.}$$

$$\cancel{E(\mathbf{w}_o)} + \delta E = \cancel{E(\mathbf{w}_o)} + (\nabla_{\mathbf{w}} E\big|_{\mathbf{w}=\mathbf{w}_o})^T \delta\mathbf{w} + \frac{1}{2!}\delta\mathbf{w}^T (\nabla^2_{\mathbf{w}} E\big|_{\mathbf{w}=\mathbf{w}_o})\delta\mathbf{w} + \mathrm{H.O.T.}$$

$$\delta E = \cancel{(\nabla_{\mathbf{w}} E\big|_{\mathbf{w}=\mathbf{w}_o})^T \delta\mathbf{w}} + \frac{1}{2!}\delta\mathbf{w}^T (\nabla^2_{\mathbf{w}} E\big|_{\mathbf{w}=\mathbf{w}_o})\delta\mathbf{w} + \cancel{\mathrm{H.O.T.}}$$

– با صرفنظرکردن از جملات درجه بالا (.H.O.T) (یعنی فرض می‌شود که معادله سطح خطا در اطراف نقطه کمینه از درجه ۲ است) و صرفنظرکردن از مشتق مرتبه اول (برای ساده‌شدن محاسبات):

پرسپترون چندلایه (MLP)

استفاده از ماتریس هِس (Hessian Matrix) سطح خطا برای هرس‌کردن شبکه:

– فرض‌کنید وزن‌ها همگرا شده باشند ($\mathbf{w}_0$) و این وزن‌ها را به‌مقدار اندکی ($\delta\mathbf{w}$) تغییر (اختلال) می‌دهیم.

– بسط تیلور تابع غیرخطی خطا حول $\mathbf{w}_0$

$$E(\mathbf{w}_o+\delta\mathbf{w}) = E(\mathbf{w}_o) + \left(\nabla_{\mathbf{w}} E\Big|_{\mathbf{w}=\mathbf{w}_o}\right)^T \delta\mathbf{w} + \frac{1}{2!}\delta\mathbf{w}^T \left(\nabla^2_{\mathbf{w}} E\Big|_{\mathbf{w}=\mathbf{w}_o}\right)\delta\mathbf{w} + \mathrm{H.O.T.}$$

$$E(\cancel{\mathbf{w}_o}) + \delta E = E(\cancel{\mathbf{w}_o}) + \left(\nabla_{\mathbf{w}} E\Big|_{\mathbf{w}=\mathbf{w}_o}\right)^T \delta\mathbf{w} + \frac{1}{2!}\delta\mathbf{w}^T \left(\nabla^2_{\mathbf{w}} E\Big|_{\mathbf{w}=\mathbf{w}_o}\right)\delta\mathbf{w} + \mathrm{H.O.T.}$$

$$\delta E = \left(\nabla_{\mathbf{w}} E\cancel{\Big|_{\mathbf{w}=\mathbf{w}_o}}\right)^T \delta\mathbf{w} + \frac{1}{2!}\delta\mathbf{w}^T \left(\nabla^2_{\mathbf{w}} E\Big|_{\mathbf{w}=\mathbf{w}_o}\right)\delta\mathbf{w} + \mathrm{H.\cancel{O}.T.}$$

– با صرفنظرکردن از جملات درجه بالا (H.O.T.) (یعنی فرض می‌شود که معادله سطح خطا در اطراف نقطه کمینه از درجه ۲ است) و صرفنظرکردن از مشتق مرتبه اول (برای ساده‌شدن محاسبات):

$$\delta E \simeq \frac{1}{2}\delta\mathbf{w}^T \mathbf{H}\,\delta\mathbf{w}$$

H: ماتریس مشتق دوم سطح خطا (ماتریس هس)

پرسپترون چندلایه (MLP)

استفاده از ماتریس هِسِ (Hessian Matrix) سطح خطا برای هرس‌کردن شبکه:

– فرض‌کنید وزن‌ها همگرا شده باشند ($\mathbf{w}_0$) و این وزن‌ها را به‌مقدار اندکی ($\delta\mathbf{w}$) تغییر (اختلال) می‌دهیم.

– بسط تیلور تابع غیرخطی خطا حول $\mathbf{w}_0$

$$E(\mathbf{w}_o + \delta\mathbf{w}) = E(\mathbf{w}_o) + (\nabla_{\mathbf{w}} E\big|_{\mathbf{w}=\mathbf{w}_o})^T \delta\mathbf{w} + \frac{1}{2!}\delta\mathbf{w}^T (\nabla^2_{\mathbf{w}} E\big|_{\mathbf{w}=\mathbf{w}_o})\,\delta\mathbf{w} + \mathrm{H.O.T.}$$

$$E(\cancel{\mathbf{w}}_o) + \delta E = E(\cancel{\mathbf{w}}_o) + (\nabla_{\mathbf{w}} E\big|_{\mathbf{w}=\mathbf{w}_o})^T \delta\mathbf{w} + \frac{1}{2!}\delta\mathbf{w}^T (\nabla^2_{\mathbf{w}} E\big|_{\mathbf{w}=\mathbf{w}_o})\,\delta\mathbf{w} + \mathrm{H.O.T.}$$

$$\delta E = (\nabla_{\mathbf{w}} E\cancel{\big|_{\mathbf{w}=\mathbf{w}_o}})^T \delta\mathbf{w} + \frac{1}{2!}\delta\mathbf{w}^T (\nabla^2_{\mathbf{w}} E\big|_{\mathbf{w}=\mathbf{w}_o})\,\delta\mathbf{w} + \mathrm{H.\cancel{O}.T.}$$

– با صرفنظرکردن از جملات درجه بالا (H.O.T.) (یعنی فرض می‌شود که معادله سطح خطا در اطراف نقطه کمینه از درجه ۲ است) و صرفنظرکردن از مشتق مرتبه اول (برای ساده‌شدن محاسبات):

$$\delta E \simeq \frac{1}{2}\delta\mathbf{w}^T \mathbf{H}\,\delta\mathbf{w}$$

H: ماتریس مشتق دوم سطح خطا (ماتریس هس)

– نتیجه: هدف، کمینه‌کردن این رابطه است؛ در شرایطی که یکی از وزن‌ها (مثلا w_i) حذف شود.

پرسپترون چندلایه (MLP)

استفاده از ماتریس هِس (Hessian Matrix) سطح خطا برای هرس‌کردن شبکه:

$$\delta E = \frac{1}{2}\delta \mathbf{w}^T \mathbf{H}\, \delta \mathbf{w}$$

پرسپترون چندلایه (MLP)

استفاده از ماتریس هِس (Hessian Matrix) سطح خطا برای هرس‌کردن شبکه:

$$\delta E = \frac{1}{2}\delta \mathbf{w}^T \mathbf{H}\, \delta \mathbf{w}$$

$$\delta w_i + w_i = 0$$

پرسپترون چندلایه (MLP)

استفاده از ماتریس هِس (Hessian Matrix) سطح خطا برای هرس‌کردن شبکه:

$$\delta E = \frac{1}{2}\delta \mathbf{w}^T \mathbf{H}\, \delta \mathbf{w}$$

$$\delta w_i + w_i = 0$$

– به‌صورت برداری

$$\mathbf{1}_i^T\, \delta \mathbf{w} + w_i = 0$$

پرسپترون چندلایه (MLP)

استفاده از ماتریس هِس (Hessian Matrix) سطح خطا برای هرس‌کردن شبکه:

$$\delta E = \frac{1}{2}\delta \mathbf{w}^T \mathbf{H}\, \delta \mathbf{w}$$

$$\delta w_i + w_i = 0$$

– به‌صورت برداری

$$\mathbf{1}_i^T\, \delta \mathbf{w} + w_i = 0$$

$$\mathbf{1}_i^T\, \delta \mathbf{w} = \begin{bmatrix} 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} \delta w_1 \\ \vdots \\ \delta w_i \\ \vdots \\ \delta w_N \end{bmatrix}$$

پرسپترون چندلایه (MLP)

استفاده از ماتریس هِسِن (Hessian Matrix) سطح خطا برای هرس‌کردن شبکه:

$$\delta E = \frac{1}{2}\delta \mathbf{w}^T \mathbf{H}\, \delta \mathbf{w} \quad (۱)$$

$$\delta w_i + w_i = 0$$

– به‌صورت برداری

$$\mathbf{1}_i^T\, \delta \mathbf{w} + w_i = 0 \quad (۲)$$

پرسپترون چندلایه (MLP)

استفاده از ماتریس هِس (Hessian Matrix) سطح خطا برای هرس‌کردن شبکه:

$$\delta E = \frac{1}{2}\delta \mathbf{w}^T \mathbf{H}\, \delta \mathbf{w} \quad (۱)$$

$$\delta w_i + w_i = 0$$

– به‌صورت برداری

$$\mathbf{1}_i^T\, \delta \mathbf{w} + w_i = 0 \quad (۲)$$

– مساله بهینه‌سازی: کمینه‌کردن (۱) با درنظرگرفتن (۲)

پرسپترون چندلایه (MLP)

استفاده از ماتریس هِس (Hessian Matrix) سطح خطا برای هرس‌کردن شبکه:

$$\delta E = \frac{1}{2}\delta \mathbf{w}^T \mathbf{H}\, \delta \mathbf{w} \quad (۱)$$

$$\delta w_i + w_i = 0$$

– به‌صورت برداری

$$\mathbf{1}_i^T\, \delta \mathbf{w} + w_i = 0 \quad (۲)$$

– مساله بهینه‌سازی: کمینه‌کردن (۱) با درنظرگرفتن (۲)

– تابع هزینه بهینه‌سازی با استفاده از ضریب لاگرانژ برابر است با:

$$S = \frac{1}{2}\delta \mathbf{w}^T \mathbf{H}\, \delta \mathbf{w} + \lambda(\mathbf{1}_i^T\, \delta \mathbf{w} + w_i)$$

پرسپترون چندلایه (MLP)

استفاده از ماتریس هِس (Hessian Matrix) سطح خطا برای هرس‌کردن شبکه:

$$\delta E = \frac{1}{2}\delta \mathbf{w}^T \mathbf{H}\, \delta \mathbf{w} \quad (۱)$$

$$\delta w_i + w_i = 0$$

– به‌صورت برداری

$$\mathbf{1}_i^T\, \delta \mathbf{w} + w_i = 0 \quad (۲)$$

– مساله بهینه‌سازی: کمینه‌کردن (۱) با درنظرگرفتن (۲)

– تابع هزینه بهینه‌سازی با استفاده از ضریب لاگرانژ برابر است با:

$$S = \frac{1}{2}\delta \mathbf{w}^T \mathbf{H}\, \delta \mathbf{w} + \lambda(\mathbf{1}_i^T\, \delta \mathbf{w} + w_i)$$

– برای حل مساله بهینه‌سازی، باید نسبت به متغیرها ($\delta \mathbf{w}$ و λ) مشتق‌گرفت و برابر صفر قرارداد:

پرسپترون چندلایه (MLP)

استفاده از ماتریس هِس (Hessian Matrix) سطح خطا برای هرس‌کردن شبکه:

$$\delta E = \frac{1}{2}\delta \mathbf{w}^T \mathbf{H}\,\delta \mathbf{w} \quad (۱)$$

$$\delta w_i + w_i = 0$$

– به‌صورت برداری

$$\mathbf{1}_i^T\,\delta \mathbf{w} + w_i = 0 \quad (۲)$$

– مساله بهینه‌سازی: کمینه‌کردن (۱) با درنظرگرفتن (۲)

– تابع هزینه بهینه‌سازی با استفاده از ضریب لاگرانژ برابر است با:

$$S = \frac{1}{2}\delta \mathbf{w}^T \mathbf{H}\,\delta \mathbf{w} + \lambda(\mathbf{1}_i^T\,\delta \mathbf{w} + w_i)$$

– برای حل مساله بهینه‌سازی، باید نسبت به متغیرها ($\delta \mathbf{w}$ و λ) مشتق‌گرفت و برابر صفر قرارداد:

$$\frac{\partial S}{\partial \delta \mathbf{w}} = \mathbf{H}\,\delta \mathbf{w} + \lambda \mathbf{1}_i^T = 0 \quad \Rightarrow \quad \delta \mathbf{w} = -\lambda\,\mathbf{H}^{-1}\mathbf{1}_i$$

پرسپترون چندلایه (MLP)

استفاده از ماتریس هِس (Hessian Matrix) سطح خطا برای هرس‌کردن شبکه:

$$\delta E = \frac{1}{2}\delta \mathbf{w}^T \mathbf{H}\, \delta \mathbf{w} \quad (۱)$$

$$\delta w_i + w_i = 0$$

– به‌صورت برداری

$$\mathbf{1}_i^T\, \delta \mathbf{w} + w_i = 0 \quad (۲)$$

– مساله بهینه‌سازی: کمینه‌کردن (۱) با درنظرگرفتن (۲)

– تابع هزینه بهینه‌سازی با استفاده از ضریب لاگرانژ برابر است با:

$$S = \frac{1}{2}\delta \mathbf{w}^T \mathbf{H}\, \delta \mathbf{w} + \lambda(\mathbf{1}_i^T\, \delta \mathbf{w} + w_i)$$

– برای حل مساله بهینه‌سازی، باید نسبت به متغیرها ($\delta \mathbf{w}$ و λ) مشتق‌گرفت و برابر صفر قرارداد:

$$\frac{\partial S}{\partial \delta \mathbf{w}} = \mathbf{H}\, \delta \mathbf{w} + \lambda \mathbf{1}_i^T = 0 \quad \Rightarrow \quad \delta \mathbf{w} = -\lambda\, \mathbf{H}^{-1} \mathbf{1}_i$$

$$\frac{\partial S}{\partial \lambda} = \mathbf{1}_i^T\, \delta \mathbf{w} + w_i = 0$$

پرسپترون چندلایه (MLP)

استفاده از ماتریس هِسِن (Hessian Matrix) سطح خطا برای هرس‌کردن شبکه:

$$\frac{\partial S}{\partial \delta \mathbf{w}} = \mathbf{H}\,\delta\mathbf{w} + \lambda \mathbf{1}_i^T = 0 \quad \Rightarrow \quad \delta\mathbf{w} = -\lambda\,\mathbf{H}^{-1}\mathbf{1}_i \quad (۱)$$

$$\frac{\partial S}{\partial \lambda} = \mathbf{1}_i^T\,\delta\mathbf{w} + w_i = 0 \quad (۲)$$

پرسپترون چندلایه (MLP)

استفاده از ماتریس هِس (Hessian Matrix) سطح خطا برای هرس‌کردن شبکه:

$$\frac{\partial S}{\partial \delta \mathbf{w}} = \mathbf{H}\,\delta\mathbf{w} + \lambda \mathbf{1}_i^T = 0 \quad \Rightarrow \quad \delta\mathbf{w} = -\lambda\,\mathbf{H}^{-1}\mathbf{1}_i \quad (۱)$$

$$\frac{\partial S}{\partial \lambda} = \mathbf{1}_i^T\,\delta\mathbf{w} + w_i = 0 \quad (۲)$$

– (۱) در (۲) نتیجه می‌دهد:

$$\mathbf{1}_i^T \lambda\,\mathbf{H}^{-1}\mathbf{1}_i = w_i \quad \Rightarrow \quad \lambda\,[\mathbf{H}^{-1}]_{ii} = w_i \quad \Rightarrow \quad \lambda = \frac{w_i}{[\mathbf{H}^{-1}]_{ii}} \quad (۳)$$

پرسپترون چندلایه (MLP)

استفاده از ماتریس هِسِ (Hessian Matrix) سطح خطا برای هرس‌کردن شبکه:

$$\frac{\partial S}{\partial \delta \mathbf{w}} = \mathbf{H}\,\delta\mathbf{w} + \lambda \mathbf{1}_i^T = 0 \quad \Rightarrow \quad \delta\mathbf{w} = -\lambda\,\mathbf{H}^{-1}\mathbf{1}_i \qquad (۱)$$

$$\frac{\partial S}{\partial \lambda} = \mathbf{1}_i^T\,\delta\mathbf{w} + w_i = 0 \qquad (۲)$$

– (۱) در (۲) نتیجه می‌دهد:

$$\mathbf{1}_i^T \lambda\,\mathbf{H}^{-1}\mathbf{1}_i = w_i \quad \Rightarrow \quad \lambda\,[\mathbf{H}^{-1}]_{ii} = w_i \quad \Rightarrow \quad \lambda = \frac{w_i}{[\mathbf{H}^{-1}]_{ii}} \qquad (۳)$$

– (۳) در (۱):

$$\delta\mathbf{w} = -\frac{w_i}{[\mathbf{H}^{-1}]_{ii}}\mathbf{H}^{-1}\mathbf{1}_i \qquad (۴)$$

پرسپترون چندلایه (MLP)

استفاده از ماتریس هِسِ (Hessian Matrix) سطح خطا برای هرس‌کردن شبکه:

$$\frac{\partial S}{\partial \delta \mathbf{w}} = \mathbf{H}\,\delta \mathbf{w} + \lambda \mathbf{1}_i^T = 0 \quad \Rightarrow \quad \delta \mathbf{w} = -\lambda\,\mathbf{H}^{-1}\mathbf{1}_i \qquad (۱)$$

$$\frac{\partial S}{\partial \lambda} = \mathbf{1}_i^T\,\delta \mathbf{w} + w_i = 0 \qquad (۲)$$

– (۱) در (۲) نتیجه می‌دهد:

$$\mathbf{1}_i^T \lambda\,\mathbf{H}^{-1}\mathbf{1}_i = w_i \quad \Rightarrow \quad \lambda\,[\mathbf{H}^{-1}]_{ii} = w_i \quad \Rightarrow \quad \lambda = \frac{w_i}{[\mathbf{H}^{-1}]_{ii}} \qquad (۳)$$

– (۳) در (۱):

$$\delta \mathbf{w} = -\frac{w_i}{[\mathbf{H}^{-1}]_{ii}}\mathbf{H}^{-1}\mathbf{1}_i \qquad (۴)$$

– در نهایت با استفاده از (۳) و (۴) درتابع هزینه S:

پرسپترون چندلایه (MLP)

استفاده از ماتریس هِس (Hessian Matrix) سطح خطا برای هرس‌کردن شبکه:

$$S_i = \frac{1}{2}\left(\frac{w_i}{[\mathbf{H}^{-1}]_{ii}}\mathbf{H}^{-1}\mathbf{1}_i\right)^T(\mathbf{H})\left(\frac{w_i}{[\mathbf{H}^{-1}]_{ii}}\mathbf{H}^{-1}\mathbf{1}_i\right) - \frac{w_i}{[\mathbf{H}^{-1}]_{ii}}\left(\mathbf{1}_i^T\frac{w_i}{[\mathbf{H}^{-1}]_{ii}}\mathbf{H}^{-1}\mathbf{1}_i - w_i\right)$$

پرسپترون چندلایه (MLP)

استفاده از ماتریس هِس (Hessian Matrix) سطح خطا برای هرس‌کردن شبکه:

$$S_i = \frac{1}{2}\left(\frac{w_i}{[\mathbf{H}^{-1}]_{ii}}\mathbf{H}^{-1}\mathbf{1}_i\right)^T(\mathbf{H})\left(\frac{w_i}{[\mathbf{H}^{-1}]_{ii}}\mathbf{H}^{-1}\mathbf{1}_i\right) - \frac{w_i}{[\mathbf{H}^{-1}]_{ii}}\left(\mathbf{1}_i^T\frac{w_i}{[\mathbf{H}^{-1}]_{ii}}\mathbf{H}^{-1}\mathbf{1}_i - w_i\right)$$

– بعد از خلاصه‌سازی (تمرین کنید):

$$S_i = \frac{1}{2}\frac{(w_i)^2}{[\mathbf{H}^{-1}]_{ii}}$$

پرسپترون چندلایه (MLP)

استفاده از ماتریس هِسِ (Hessian Matrix) سطح خطا برای هرس‌کردن شبکه:

$$S_i = \frac{1}{2}\left(\frac{w_i}{[\mathbf{H}^{-1}]_{ii}}\mathbf{H}^{-1}\mathbf{1}_i\right)^T(\mathbf{H})\left(\frac{w_i}{[\mathbf{H}^{-1}]_{ii}}\mathbf{H}^{-1}\mathbf{1}_i\right) - \frac{w_i}{[\mathbf{H}^{-1}]_{ii}}\left(\mathbf{1}_i^T\frac{w_i}{[\mathbf{H}^{-1}]_{ii}}\mathbf{H}^{-1}\mathbf{1}_i - w_i\right)$$

– بعد از خلاصه‌سازی (تمرین کنید):

$$S_i = \frac{1}{2}\frac{(w_i)^2}{[\mathbf{H}^{-1}]_{ii}}$$

۱– این رابطه بیان می‌کند که در اثر حذف وزن w_i مقدار خطای مدل‌سازی چقدر افزایش می‌یابد.

پرسپترون چندلایه (MLP)

استفاده از ماتریس هِس (Hessian Matrix) سطح خطا برای هرس‌کردن شبکه:

$$S_i = \frac{1}{2}\left(\frac{w_i}{[\mathbf{H}^{-1}]_{ii}}\mathbf{H}^{-1}\mathbf{1}_i\right)^T(\mathbf{H})\left(\frac{w_i}{[\mathbf{H}^{-1}]_{ii}}\mathbf{H}^{-1}\mathbf{1}_i\right) - \frac{w_i}{[\mathbf{H}^{-1}]_{ii}}\left(\mathbf{1}_i^T\frac{w_i}{[\mathbf{H}^{-1}]_{ii}}\mathbf{H}^{-1}\mathbf{1}_i - w_i\right)$$

- بعد از خلاصه‌سازی (تمرین کنید):

$$S_i = \frac{1}{2}\frac{(w_i)^2}{[\mathbf{H}^{-1}]_{ii}}$$

۱- این رابطه بیان می‌کند که در اثر حذف وزن w_i مقدار خطای مدل‌سازی چقدر افزایش می‌یابد.

۲- عیب عمده این روش، محاسبه ماتریس هِس و به‌دست‌آوردن وارون آن است.

پرسپترون چندلایه (MLP)

روش نیوتن برای تنظیم وزن‌های شبکه:

پرسپترون چندلایه (MLP)

روش نیوتن برای تنظیم وزن‌های شبکه:

– رابطه قبلی در روش هرس‌کردن درنظر بگیرید:

$$\delta E = \left(\nabla_{\mathbf{w}} E \Big|_{\mathbf{w}=\mathbf{w}_o}\right)^T \delta\mathbf{w} + \frac{1}{2!}\delta\mathbf{w}^T \left(\nabla^2_{\mathbf{w}} E \Big|_{\mathbf{w}=\mathbf{w}_o}\right)\delta\mathbf{w} + \mathrm{H.O.T.}$$

پرسپترون چندلایه (MLP)

روش نیوتن برای تنظیم وزن‌های شبکه:

– رابطه قبلی در روش هرس‌کردن درنظر بگیرید:

$$\delta E = \left(\nabla_{\mathbf{w}} E \Big|_{\mathbf{w}=\mathbf{w}_o}\right)^T \delta\mathbf{w} + \frac{1}{2!}\delta\mathbf{w}^T \left(\nabla^2_{\mathbf{w}} E \Big|_{\mathbf{w}=\mathbf{w}_o}\right) \delta\mathbf{w} + \mathrm{H.O.T.}$$

– با صرفنظرکردن از H.O.T:

$$\delta E \simeq \mathbf{g}^T \delta\mathbf{w} + \frac{1}{2}\delta\mathbf{w}^T \mathbf{H}\,\delta\mathbf{w}$$

پرسپترون چندلایه (MLP)

<u>روش نیوتن برای تنظیم وزن‌های شبکه:</u>

– رابطه قبلی در روش هرس‌کردن درنظر بگیرید:

$$\delta E = \left(\nabla_{\mathbf{w}} E\Big|_{\mathbf{w}=\mathbf{w}_o}\right)^T \delta\mathbf{w} + \frac{1}{2!}\delta\mathbf{w}^T \left(\nabla^2_{\mathbf{w}} E\Big|_{\mathbf{w}=\mathbf{w}_o}\right)\delta\mathbf{w} + \text{H.O.T.}$$

– با صرفنظرکردن از H.O.T:

$$\delta E \simeq \mathbf{g}^T\delta\mathbf{w} + \frac{1}{2}\delta\mathbf{w}^T\mathbf{H}\,\delta\mathbf{w}$$

– برای کمینه‌کردن این تابع هزینه باید از آن نسبت به متغیر آن مشتق گرفته و برابر صفر قرارداد

$$\frac{\partial(\delta E)}{\partial(\delta\mathbf{w})} = \mathbf{g} + \mathbf{H}\,\delta\mathbf{w} = 0 \quad\Rightarrow\quad \delta\mathbf{w} = -\mathbf{H}^{-1}\mathbf{g}$$

پرسپترون چندلایه (MLP)

روش نیوتن برای تنظیم وزن‌های شبکه:

– رابطه قبلی در روش هرس‌کردن درنظر بگیرید:

$$\delta E = \left(\nabla_{\mathbf{w}} E\Big|_{\mathbf{w}=\mathbf{w}_o}\right)^T \delta\mathbf{w} + \frac{1}{2!}\delta\mathbf{w}^T\left(\nabla^2_{\mathbf{w}} E\Big|_{\mathbf{w}=\mathbf{w}_o}\right)\delta\mathbf{w} + \mathrm{H.O.T.}$$

– با صرفنظرکردن از H.O.T:

$$\delta E \simeq \mathbf{g}^T\delta\mathbf{w} + \frac{1}{2}\delta\mathbf{w}^T\mathbf{H}\,\delta\mathbf{w}$$

– برای کمینه‌کردن این تابع هزینه باید از آن نسبت به متغیر آن مشتق گرفته و برابر صفر قرارداد

$$\frac{\partial(\delta E)}{\partial(\delta\mathbf{w})} = \mathbf{g} + \mathbf{H}\,\delta\mathbf{w} = 0 \quad\Rightarrow\quad \delta\mathbf{w} = -\mathbf{H}^{-1}\mathbf{g}$$

– و یا

$$\Delta\,\mathbf{w} = -\mathbf{H}^{-1}\mathbf{g}$$

پرسپترون چندلایه (MLP)

روش نیوتن برای تنظیم وزن‌های شبکه:

پرسپترون چندلایه (MLP)

روش نیوتن برای تنظیم وزن‌های شبکه:

– رابطه تنظیم وزن‌ها به‌صورت تکراری

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \Delta\mathbf{w}(n)$$

پرسپترون چندلایه (MLP)

روش نیوتن برای تنظیم وزن‌های شبکه:

- رابطه تنظیم وزن‌ها به‌صورت تکراری

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \Delta\mathbf{w}(n)$$

- در نهایت

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mathbf{H}^{-1}(n)\,\mathbf{g}(n)$$

پرسپترون چندلایه (MLP)

روش نیوتن برای تنظیم وزن‌های شبکه:

– رابطه تنظیم وزن‌ها به‌صورت تکراری

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \Delta\mathbf{w}(n)$$

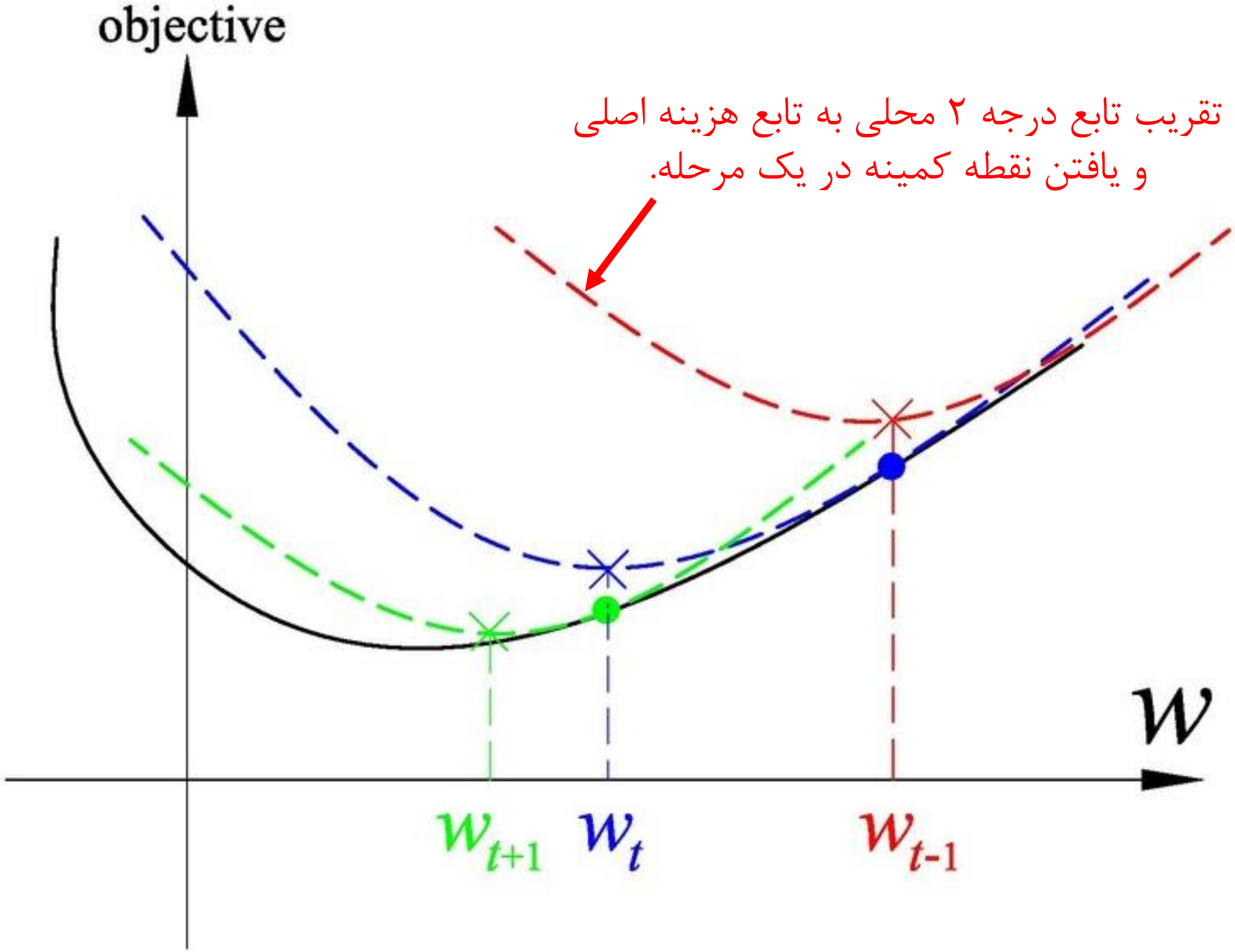
– در نهایت

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mathbf{H}^{-1}(n)\,\mathbf{g}(n)$$

– این معادله تنظیم وزن‌ها نیز همانند الگوریتم پس‌انتشار خطا به‌صورت تکراری ادامه یافته تا خطا همگرا شود.

پرسپترون چندلایه (MLP)

روش نیوتن برای تنظیم وزن‌های شبکه:



پرسپترون چندلایه (MLP)

روش نیوتن برای تنظیم وزن‌های شبکه:

پرسپترون چندلایه (MLP)

روش نیوتن برای تنظیم وزن‌های شبکه:

– اگرچه روش نیوتن، روشی بسیار سریع برای حل مساله‌های بهینه‌سازی است، معایبی نیز دارد:

پرسپترون چندلایه (MLP)

روش نیوتن برای تنظیم وزن‌های شبکه:

– اگرچه روش نیوتن، روشی بسیار سریع برای حل مساله‌های بهینه‌سازی است، معایبی نیز دارد:

- نیاز به محاسبه وارون ماتریس هس دارد که

پرسپترون چندلایه (MLP)

روش نیوتن برای تنظیم وزن‌های شبکه:

– اگرچه روش نیوتن، روشی بسیار سریع برای حل مساله‌های بهینه‌سازی است، معایبی نیز دارد:

- نیاز به محاسبه وارون ماتریس هس دارد که
  ۱- هزینه محاسباتی دارد

پرسپترون چندلایه (MLP)

روش نیوتن برای تنظیم وزن‌های شبکه:

- اگرچه روش نیوتن، روشی بسیار سریع برای حل مساله‌های بهینه‌سازی است، معایبی نیز دارد:

• نیاز به محاسبه وارون ماتریس هس دارد که

۱- هزینه محاسباتی دارد

۲- ممکن است تکین باشد.

پرسپترون چندلایه (MLP)

<u>روش نیوتن برای تنظیم وزن‌های شبکه:</u>

– اگرچه روش نیوتن، روشی بسیار سریع برای حل مساله‌های بهینه‌سازی است، معایبی نیز دارد:

- نیاز به محاسبه وارون ماتریس هس دارد که
  ۱- هزینه محاسباتی دارد
  ۲- ممکن است تکین باشد.
- هنگامی که تابع هزینه نسبت به وزن‌های شبکه از مرتبه ۲ نباشد، همگرایی روش نیوتن با مشکل مواجه می‌شود.

پرسپترون چندلایه (MLP)

روش نیوتن برای تنظیم وزن‌های شبکه:

– اگرچه روش نیوتن، روشی بسیار سریع برای حل مساله‌های بهینه‌سازی است، معایبی نیز دارد:

- نیاز به محاسبه وارون ماتریس هس دارد که
  - ۱– هزینه محاسباتی دارد
  - ۲– ممکن است تکین باشد.
- هنگامی که تابع هزینه نسبت به وزن‌های شبکه از مرتبه ۲ نباشد، همگرایی روش نیوتن با مشکل مواجه می‌شود.

– بنابراین، از این روش در آموزش شبکه‌های عصبی استفاده نمی‌شود.

پرسپترون چندلایه (MLP)

روش نیوتن برای تنظیم وزن‌های شبکه:

- اگرچه روش نیوتن، روشی بسیار سریع برای حل مساله‌های بهینه‌سازی است، معایبی نیز دارد:

  • نیاز به محاسبه وارون ماتریس هس دارد که
    ۱- هزینه محاسباتی دارد
    ۲- ممکن است تکین باشد.

  • هنگامی که تابع هزینه نسبت به وزن‌های شبکه از مرتبه ۲ نباشد، همگرایی روش نیوتن با مشکل مواجه می‌شود.

- بنابراین، از این روش در آموزش شبکه‌های عصبی استفاده نمی‌شود.

- روش‌های شبه نیوتن (quasi-Newton) و کاذب نیوتن (pseudo-Newton) مناسب‌تر هستند.

پرسپترون چندلایه (MLP)

روش نیوتن برای تنظیم وزن‌های شبکه:

– اگرچه روش نیوتن، روشی بسیار سریع برای حل مساله‌های بهینه‌سازی است، معایبی نیز دارد:

- نیاز به محاسبه وارون ماتریس هس دارد که
  - ۱– هزینه محاسباتی دارد
  - ۲– ممکن است تکین باشد.
- هنگامی که تابع هزینه نسبت به وزن‌های شبکه از مرتبه ۲ نباشد، همگرایی روش نیوتن با مشکل مواجه می‌شود.

– بنابراین، از این روش در آموزش شبکه‌های عصبی استفاده نمی‌شود.

– روش‌های شبه نیوتن (quasi-Newton) و کاذب نیوتن (pseudo-Newton) مناسب‌تر هستند.

– مناسب‌ترین روش مرتبه ۲، روش لونبرگ–مارکارت (Levenberg-Marquardt) است.

پرسپترون چندلایه (MLP)

روش لونبرگ-مارکارت (Levenberg-Marquardt):

پرسپترون چندلایه (MLP)

<u>روش لونبرگ-مارکارت (Levenberg-Marquardt):</u>

- این روش، مصالحه‌ای است بین:

پرسپترون چندلایه (MLP)

روش لونبرگ-مارکارت (Levenberg-Marquardt):

- **این روش، مصالحه‌ای است بین:**
  - **روش نیوتن که روشی بسیار سریع است ولی ممکن است واگرا شود.**

پرسپترون چندلایه (MLP)

روش لونبرگ-مارکارت (Levenberg-Marquardt):

- این روش، مصالحه‌ای است بین:
  - روش نیوتن که روشی بسیار سریع است ولی ممکن است واگرا شود.
  - روش گرادیان نزولی که همگرایی آن با انتخاب مناسب ضریب آموزش تضمین می‌شود ولی به آهستگی همگرا می شود.

پرسپترون چندلایه (MLP)

روش لونبرگ–مارکارت (Levenberg-Marquardt):

– این روش، مصالحه‌ای است بین:

- روش نیوتن که روشی بسیار سریع است ولی ممکن است واگرا شود.
- روش گرادیان نزولی که همگرایی آن با انتخاب مناسب ضریب آموزش تضمین می‌شود ولی به آهستگی همگرا می شود.

– دو روش آموزش گرادیان نزولی (پس‌انتشار خطا) و نیوتن را به‌خاطر بیاورید:

$$\Delta \mathbf{w} = -\eta \mathbf{g}$$

$$\Delta \mathbf{w} = -\mathbf{H}^{-1} \mathbf{g}$$

پرسپترون چندلایه (MLP)

روش لونبرگ-مارکارت (Levenberg-Marquardt):

– این روش، مصالحه‌ای است بین:

- روش نیوتن که روشی بسیار سریع است ولی ممکن است واگرا شود.
- روش گرادیان نزولی که همگرایی آن با انتخاب مناسب ضریب آموزش تضمین می‌شود ولی به آهستگی همگرا می شود.

– دو روش آموزش گرادیان نزولی (پس‌انتشار خطا) و نیوتن را به‌خاطر بیاورید:

$$\Delta \mathbf{w} = -\eta \mathbf{g}$$
$$\Delta \mathbf{w} = -\mathbf{H}^{-1}\mathbf{g} \quad \Rightarrow \quad \Delta \mathbf{w} = -(\mathbf{H} + \lambda \mathbf{I})^{-1}\mathbf{g}$$

پرسپترون چندلایه (MLP)

روش لونبرگ-مارکارت (Levenberg-Marquardt):

– این روش، مصالحه‌ای است بین:

- روش نیوتن که روشی بسیار سریع است ولی ممکن است واگرا شود.
- روش گرادیان نزولی که همگرایی آن با انتخاب مناسب ضریب آموزش تضمین می‌شود ولی به آهستگی همگرا می شود.

– دو روش آموزش گرادیان نزولی (پس‌انتشار خطا) و نیوتن را به‌خاطر بیاورید:

$$\begin{aligned} \Delta \mathbf{w} &= -\eta \mathbf{g} \\ \Delta \mathbf{w} &= -\mathbf{H}^{-1}\mathbf{g} \end{aligned} \quad \Rightarrow \quad \Delta \mathbf{w} = -(\mathbf{H} + \lambda \mathbf{I})^{-1}\mathbf{g}$$

– نتیجه‌گیری:

پرسپترون چندلایه (MLP)

روش لونبرگ-مارکارت (Levenberg-Marquardt):

– این روش، مصالحه‌ای است بین:

- روش نیوتن که روشی بسیار سریع است ولی ممکن است واگرا شود.
- روش گرادیان نزولی که همگرایی آن با انتخاب مناسب ضریب آموزش تضمین می‌شود ولی به آهستگی همگرا می شود.

– دو روش آموزش گرادیان نزولی (پس‌انتشار خطا) و نیوتن را به‌خاطر بیاورید:

$$\begin{aligned} \Delta \mathbf{w} &= -\eta \mathbf{g} \\ \Delta \mathbf{w} &= -\mathbf{H}^{-1}\mathbf{g} \end{aligned} \quad \Longrightarrow \quad \Delta \mathbf{w} = -(\mathbf{H} + \lambda \mathbf{I})^{-1}\mathbf{g}$$

– نتیجه‌گیری:

۱- با انتخاب مناسب ضریب λ ، وارون ماتریس $\mathbf{H}$ همواره وجود خواهدداشت (روش تنظیم‌کننده).

پرسپترون چندلایه (MLP)

روش لونبرگ-مارکارت (Levenberg-Marquardt):

- این روش، مصالحه‌ای است بین:
  - روش نیوتن که روشی بسیار سریع است ولی ممکن است واگرا شود.
  - روش گرادیان نزولی که همگرایی آن با انتخاب مناسب ضریب آموزش تضمین می‌شود ولی به آهستگی همگرا می شود.

- دو روش آموزش گرادیان نزولی (پس‌انتشار خطا) و نیوتن را به‌خاطر بیاورید:

$$\begin{array}{l} \Delta\,\mathbf{w} = -\eta\mathbf{g} \\ \Delta\,\mathbf{w} = -\mathbf{H}^{-1}\mathbf{g} \end{array} \quad \Longrightarrow \quad \Delta\,\mathbf{w} = -(\mathbf{H} + \lambda\mathbf{I})^{-1}\mathbf{g}$$

- نتیجه‌گیری:

۱- با انتخاب مناسب ضریب λ ، وارون ماتریس $\mathbf{H}$ همواره وجود خواهدداشت (روش تنظیم‌کننده).

۲- برای $\lambda \ll 1$ $\Leftarrow$ ؟

پرسپترون چندلایه (MLP)

روش لونبرگ-مارکارت (Levenberg-Marquardt):

– این روش، مصالحه‌ای است بین:

- روش نیوتن که روشی بسیار سریع است ولی ممکن است واگرا شود.
- روش گرادیان نزولی که همگرایی آن با انتخاب مناسب ضریب آموزش تضمین می‌شود ولی به آهستگی همگرا می شود.

– دو روش آموزش گرادیان نزولی (پس‌انتشار خطا) و نیوتن را به‌خاطر بیاورید:

$$\begin{array}{l}\Delta \mathbf{w} = -\eta \mathbf{g} \\ \Delta \mathbf{w} = -\mathbf{H}^{-1}\mathbf{g}\end{array} \quad \Longrightarrow \quad \Delta \mathbf{w} = -(\mathbf{H} + \lambda \mathbf{I})^{-1}\mathbf{g}$$

– نتیجه‌گیری:

۱- با انتخاب مناسب ضریب λ ، وارون ماتریس **H** همواره وجود خواهدداشت (روش تنظیم‌کننده).

۲- برای $\lambda \ll 1$ $\Leftarrow$ ؟ روش نیوتن

پرسپترون چندلایه (MLP)

روش لونبرگ-مارکارت (Levenberg-Marquardt):

- این روش، مصالحه‌ای است بین:
  - روش نیوتن که روشی بسیار سریع است ولی ممکن است واگرا شود.
  - روش گرادیان نزولی که همگرایی آن با انتخاب مناسب ضریب آموزش تضمین می‌شود ولی به آهستگی همگرا می شود.

- دو روش آموزش گرادیان نزولی (پس‌انتشار خطا) و نیوتن را به‌خاطر بیاورید:

$$\begin{aligned} \Delta \mathbf{w} &= -\eta \mathbf{g} \\ \Delta \mathbf{w} &= -\mathbf{H}^{-1}\mathbf{g} \end{aligned} \quad \Longrightarrow \quad \Delta \mathbf{w} = -(\mathbf{H} + \lambda \mathbf{I})^{-1}\mathbf{g}$$

- نتیجه‌گیری:

۱- با انتخاب مناسب ضریب λ ، وارون ماتریس $\mathbf{H}$ همواره وجود خواهدداشت (روش تنظیم‌کننده).

۲- برای $1 \ll \lambda$ $\Leftarrow$ ؟ روش نیوتن

۳- برای $1 \gg \lambda$ $\Leftarrow$ ؟

پرسپترون چندلایه (MLP)

روش لونبرگ-مارکارت (Levenberg-Marquardt):

– این روش، مصالحه‌ای است بین:

- روش نیوتن که روشی بسیار سریع است ولی ممکن است واگرا شود.
- روش گرادیان نزولی که همگرایی آن با انتخاب مناسب ضریب آموزش تضمین می‌شود ولی به آهستگی همگرا می شود.

– دو روش آموزش گرادیان نزولی (پس‌انتشار خطا) و نیوتن را به‌خاطر بیاورید:

$$\Delta \mathbf{w} = -\eta \mathbf{g}$$
$$\Delta \mathbf{w} = -\mathbf{H}^{-1}\mathbf{g} \quad \Longrightarrow \quad \Delta \mathbf{w} = -(\mathbf{H} + \lambda \mathbf{I})^{-1}\mathbf{g}$$

– نتیجه‌گیری:

۱- با انتخاب مناسب ضریب λ ، وارون ماتریس $\mathbf{H}$ همواره وجود خواهدداشت (روش تنظیم‌کننده).

۲- برای $\lambda \ll 1$ $\Leftarrow$ ؟ روش نیوتن

۳- برای $\lambda \gg 1$ $\Leftarrow$ ؟ روش گرادیان نزولی

پرسپترون چندلایه (MLP)

روش لونبرگ-مارکارت (Levenberg-Marquardt):

پرسپترون چندلایه (MLP)

روش لونبرگ-مارکارت (Levenberg-Marquardt):

- **جزییات بیشتر در مورد این روش:**

پرسپترون چندلایه (MLP)

روش لونبرگ-مارکارت (Levenberg-Marquardt):

- **جزییات بیشتر در مورد این روش:**
- **فرض کنید خروجی شبکه MLP توسط تابع $F(\mathbf{w})$ نشان داده شود.**

پرسپترون چندلایه (MLP)

روش لونبرگ-مارکارت (Levenberg-Marquardt):

– جزییات بیشتر در مورد این روش:

– فرض کنید خروجی شبکه **MLP** توسط تابع $F(\mathbf{w})$ نشان داده شود.

– در این‌صورت خطای مدل‌سازی برای شبکه‌ای با یک خروجی:

$$E(n) = \frac{1}{2}\left[d(n) - F(\mathbf{w}(n))\right]^2$$

پرسپترون چندلایه (MLP)

روش لونبرگ-مارکارت (Levenberg-Marquardt):

– جزییات بیشتر در مورد این روش:

– فرض کنید خروجی شبکه MLP توسط تابع $F(\mathbf{w})$ نشان داده شود.

– در این‌صورت خطای مدل‌سازی برای شبکه‌ای با یک خروجی:

$$E(n) = \frac{1}{2}\left[d(n) - F(\mathbf{w}(n))\right]^2$$

– بردار گرادیان (مشتق مرتبه اول) و ماتریس هِس (مشتق مرتبه دوم) این خطا:

پرسپترون چندلایه (MLP)

روش لونبرگ-مارکارت (Levenberg-Marquardt):

– جزییات بیشتر در مورد این روش:

– فرض کنید خروجی شبکه MLP توسط تابع $F(\mathbf{w})$ نشان داده شود.

– در این‌صورت خطای مدل‌سازی برای شبکه‌ای با یک خروجی:

$$E(n) = \frac{1}{2}\left[d(n) - F(\mathbf{w}(n))\right]^2$$

– بردار گرادیان (مشتق مرتبه اول) و ماتریس هِس (مشتق مرتبه دوم) این خطا:

$$\mathbf{g}(\mathbf{w}(n)) = \frac{\partial E(n)}{\partial \mathbf{w}(n)} = -\left[d(n) - F(\mathbf{w}(n))\right]\frac{\partial F(\mathbf{w}(n))}{\partial \mathbf{w}(n)}$$

پرسپترون چندلایه (MLP)

روش لونبرگ-مارکارت (Levenberg-Marquardt):

– جزییات بیشتر در مورد این روش:

– فرض کنید خروجی شبکه **MLP** توسط تابع $F(\mathbf{w})$ نشان داده شود.

– در این‌صورت خطای مدل‌سازی برای شبکه‌ای با یک خروجی:

$$E(n) = \frac{1}{2}\left[d(n) - F(\mathbf{w}(n))\right]^2$$

– بردار گرادیان (مشتق مرتبه اول) و ماتریس هِس (مشتق مرتبه دوم) این خطا:

$$\mathbf{g}(\mathbf{w}(n)) = \frac{\partial E(n)}{\partial \mathbf{w}(n)} = -\left[d(n) - F(\mathbf{w}(n))\right]\frac{\partial F(\mathbf{w}(n))}{\partial \mathbf{w}(n)}$$

$$\mathbf{H}(\mathbf{w}(n)) = \frac{\partial^2 E(n)}{\partial \mathbf{w}^2(n)} = \left[\frac{\partial F(\mathbf{w}(n))}{\partial \mathbf{w}}\right]\left[\frac{\partial F(\mathbf{w}(n))}{\partial \mathbf{w}}\right]^T - \left[d(n) - F(\mathbf{w}(n))\right]\frac{\partial^2 F(\mathbf{w}(n))}{\partial \mathbf{w}^2}$$

پرسپترون چندلایه (MLP)

روش لونبرگ-مارکارت (Levenberg-Marquardt):

– جزییات بیشتر در مورد این روش:

– فرض کنید خروجی شبکه **MLP** توسط تابع $F(\mathbf{w})$ نشان داده شود.

– در این‌صورت خطای مدل‌سازی برای شبکه‌ای با یک خروجی:

$$E(n) = \frac{1}{2}\left[d(n) - F(\mathbf{w}(n))\right]^2$$

– بردار گرادیان (مشتق مرتبه اول) و ماتریس هِس (مشتق مرتبه دوم) این خطا:

$$\mathbf{g}(\mathbf{w}(n)) = \frac{\partial E(n)}{\partial \mathbf{w}(n)} = -\left[d(n) - F(\mathbf{w}(n))\right]\frac{\partial F(\mathbf{w}(n))}{\partial \mathbf{w}(n)}$$

$$\mathbf{H}(\mathbf{w}(n)) = \frac{\partial^2 E(n)}{\partial \mathbf{w}^2(n)} = \left[\frac{\partial F(\mathbf{w}(n))}{\partial \mathbf{w}}\right]\left[\frac{\partial F(\mathbf{w}(n))}{\partial \mathbf{w}}\right]^T - \left[d(n) - F(\mathbf{w}(n))\right]\frac{\partial^2 F(\mathbf{w}(n))}{\partial \mathbf{w}^2}$$

– توجه‌کنید که محاسبه ماتریس **H** بسیار زمانبر است، مخصوصا هنگامی که تعداد وزن‌ها زیاد باشد.

پرسپترون چندلایه (MLP)

روش لونبرگ-مارکارت (Levenberg-Marquardt):

$$\mathbf{H}(\mathbf{w}(n)) = \frac{\partial^2 E(n)}{\partial \mathbf{w}^2(n)} = \left[\frac{\partial F(\mathbf{w}(n))}{\partial \mathbf{w}}\right]\left[\frac{\partial F(\mathbf{w}(n))}{\partial \mathbf{w}}\right]^T - \left[d(n) - F(\mathbf{w}(n))\right]\frac{\partial^2 F(\mathbf{w}(n))}{\partial \mathbf{w}^2}$$

پرسپترون چندلایه (MLP)

روش لونبرگ-مارکارت (Levenberg-Marquardt):

$$\mathbf{H}(\mathbf{w}(n)) = \frac{\partial^2 E(n)}{\partial \mathbf{w}^2(n)} = \left[\frac{\partial F(\mathbf{w}(n))}{\partial \mathbf{w}}\right]\left[\frac{\partial F(\mathbf{w}(n))}{\partial \mathbf{w}}\right]^T \cancel{- \left[d(n) - F(\mathbf{w}(n))\right]\frac{\partial^2 F(\mathbf{w}(n))}{\partial \mathbf{w}^2}}$$

– برای ساده‌ترشدن محاسبات، می‌توان از جمله دوم صرفنظر کرد. البته در حوالی نقطه کمینه، این کار توجیه‌پذیر است زیرا خطا مقدار بسیار کوچکی دارد.

پرسپترون چندلایه (MLP)

روش لونبرگ-مارکارت (Levenberg-Marquardt):

$$\mathbf{H}(\mathbf{w}(n)) = \frac{\partial^2 E(n)}{\partial \mathbf{w}^2(n)} = \left[\frac{\partial F(\mathbf{w}(n))}{\partial \mathbf{w}}\right]\left[\frac{\partial F(\mathbf{w}(n))}{\partial \mathbf{w}}\right]^T \cancel{- \left[d(n) - F(\mathbf{w}(n))\right]\frac{\partial^2 F(\mathbf{w}(n))}{\partial \mathbf{w}^2}}$$

– برای ساده‌ترشدن محاسبات، می‌توان از جمله دوم صرفنظر کرد. البته در حوالی نقطه کمینه، این کار توجیه‌پذیر است زیرا خطا مقدار بسیار کوچکی دارد.

– البته باید توجه داشت که این کار همان بهینه‌سازی مرتبه اول است که در آن از بردار گرادیان و ضرب خارجی آن در تنظیم وزن‌ها استفاده شده است.

پرسپترون چندلایه (MLP)

روش لونبرگ-مارکارت (Levenberg-Marquardt):

$$\mathbf{H}(\mathbf{w}(n)) = \frac{\partial^2 E(n)}{\partial \mathbf{w}^2(n)} = \left[\frac{\partial F(\mathbf{w}(n))}{\partial \mathbf{w}}\right]\left[\frac{\partial F(\mathbf{w}(n))}{\partial \mathbf{w}}\right]^T \cancel{- \left[d(n) - F(\mathbf{w}(n))\right]\frac{\partial^2 F(\mathbf{w}(n))}{\partial \mathbf{w}^2}}$$

- برای ساده‌ترشدن محاسبات، می‌توان از جمله دوم صرفنظر کرد. البته در حوالی نقطه کمینه، این کار توجیه‌پذیر است زیرا خطا مقدار بسیار کوچکی دارد.

- البته باید توجه داشت که این کار همان بهینه‌سازی مرتبه اول است که در آن از بردار گرادیان و ضرب خارجی آن در تنظیم وزن‌ها استفاده شده است.

- چنانچه تعداد وزن‌ها زیاد نباشد، مناسب است که جمله دوم در رابطه بالا حذف نشود.

پرسپترون چندلایه (MLP)

روش لونبرگ-مارکارت (Levenberg-Marquardt):

- الگوریتم لونبرگ-مارکارت:

پرسپترون چندلایه (MLP)

روش لونبرگ-مارکارت (Levenberg-Marquardt):

- **الگوریتم لونبرگ-مارکارت:**

۱- ذخیره نگه‌داشتن $E(\mathbf{w}(n-1))$ از مرحله قبلی

پرسپترون چندلایه (MLP)

روش لونبرگ-مارکارت (Levenberg-Marquardt):

- الگوریتم لونبرگ-مارکارت:

۱- ذخیره نگه‌داشتن $E(\mathbf{w}(n-1))$ از مرحله قبلی

۲- انتخاب مقداری نسبتا کم برای λ (مثلا $\lambda = 10^{-3}$)

پرسپترون چندلایه (MLP)

روش لونبرگ-مارکارت (Levenberg-Marquardt):

- الگوریتم لونبرگ-مارکارت:

۱- ذخیره نگه‌داشتن $E(\mathbf{w}(n-1))$ از مرحله قبلی

۲- انتخاب مقداری نسبتا کم برای λ (مثلا $\lambda = 10^{-3}$)

۳- محاسبه بردار گرادیان، ماتریس هِس، و تنظیم وزن‌ها

$$\Delta\mathbf{w}(n) = -\left(\mathbf{H}(n) + \lambda\mathbf{I}\right)^{-1}\mathbf{g}(n)$$

پرسپترون چندلایه (MLP)

روش لونبرگ-مارکارت (Levenberg-Marquardt):

- الگوریتم لونبرگ-مارکارت:

۱- ذخیره نگه‌داشتن $E(\mathbf{w}(n-1))$ از مرحله قبلی

۲- انتخاب مقداری نسبتا کم برای λ (مثلا $\lambda = 10^{-3}$)

۳- محاسبه بردار گرادیان، ماتریس هِس، و تنظیم وزن‌ها

$$\Delta\mathbf{w}(n) = -\left(\mathbf{H}(n) + \lambda\mathbf{I}\right)^{-1}\mathbf{g}(n)$$

۴- محاسبه $E\left(\mathbf{w}(n) + \Delta\mathbf{w}(n)\right)$:

پرسپترون چندلایه (MLP)

روش لونبرگ-مارکارت (Levenberg-Marquardt):

- الگوریتم لونبرگ-مارکارت:

۱- ذخیره نگه‌داشتن $E(\mathbf{w}(n-1))$ از مرحله قبلی

۲- انتخاب مقداری نسبتا کم برای λ (مثلا $\lambda = 10^{-3}$)

۳- محاسبه بردار گرادیان، ماتریس هِس، و تنظیم وزن‌ها

$$\Delta\mathbf{w}(n) = -\left(\mathbf{H}(n) + \lambda\mathbf{I}\right)^{-1}\mathbf{g}(n)$$

۴- محاسبه $E\left(\mathbf{w}(n) + \Delta\mathbf{w}(n)\right)$:

آ- چنانچه $E\left(\mathbf{w}(n)+\Delta\mathbf{w}(n)\right) \geq E\left(\mathbf{w}(n-1)\right)$ ، λ ده برابر شود. بازگشت به گام ۳.

پرسپترون چندلایه (MLP)

روش لونبرگ-مارکارت (Levenberg-Marquardt):

- الگوریتم لونبرگ-مارکارت:

۱- ذخیره نگه‌داشتن $E(\mathbf{w}(n-1))$ از مرحله قبلی

۲- انتخاب مقداری نسبتا کم برای λ (مثلا $\lambda = 10^{-3}$)

۳- محاسبه بردار گرادیان، ماتریس هِس، و تنظیم وزن‌ها

$$\Delta\mathbf{w}(n) = -\left(\mathbf{H}(n) + \lambda\mathbf{I}\right)^{-1}\mathbf{g}(n)$$

۴- محاسبه $E\left(\mathbf{w}(n) + \Delta\mathbf{w}(n)\right)$:

آ- چنانچه $E\left(\mathbf{w}(n) + \Delta\mathbf{w}(n)\right) \geq E\left(\mathbf{w}(n-1)\right)$ ، λ ده برابر شود. بازگشت به گام ۳.

ب- چنانچه $E\left(\mathbf{w}(n) + \Delta\mathbf{w}(n)\right) < E\left(\mathbf{w}(n-1)\right)$ ، λ یک دهم شود. بازگشت به گام ۳.

پرسپترون چندلایه (MLP)

روش لونبرگ-مارکارت (Levenberg-Marquardt):

– الگوریتم لونبرگ-مارکارت:

۱– ذخیره نگه‌داشتن $E(\mathbf{w}(n-1))$ از مرحله قبلی

۲– انتخاب مقداری نسبتا کم برای λ (مثلا $\lambda = 10^{-3}$)

۳– محاسبه بردار گرادیان، ماتریس هِس، و تنظیم وزن‌ها

$$\Delta\mathbf{w}(n) = -\left(\mathbf{H}(n) + \lambda\mathbf{I}\right)^{-1}\mathbf{g}(n)$$

۴– محاسبه $E\left(\mathbf{w}(n) + \Delta\mathbf{w}(n)\right)$:

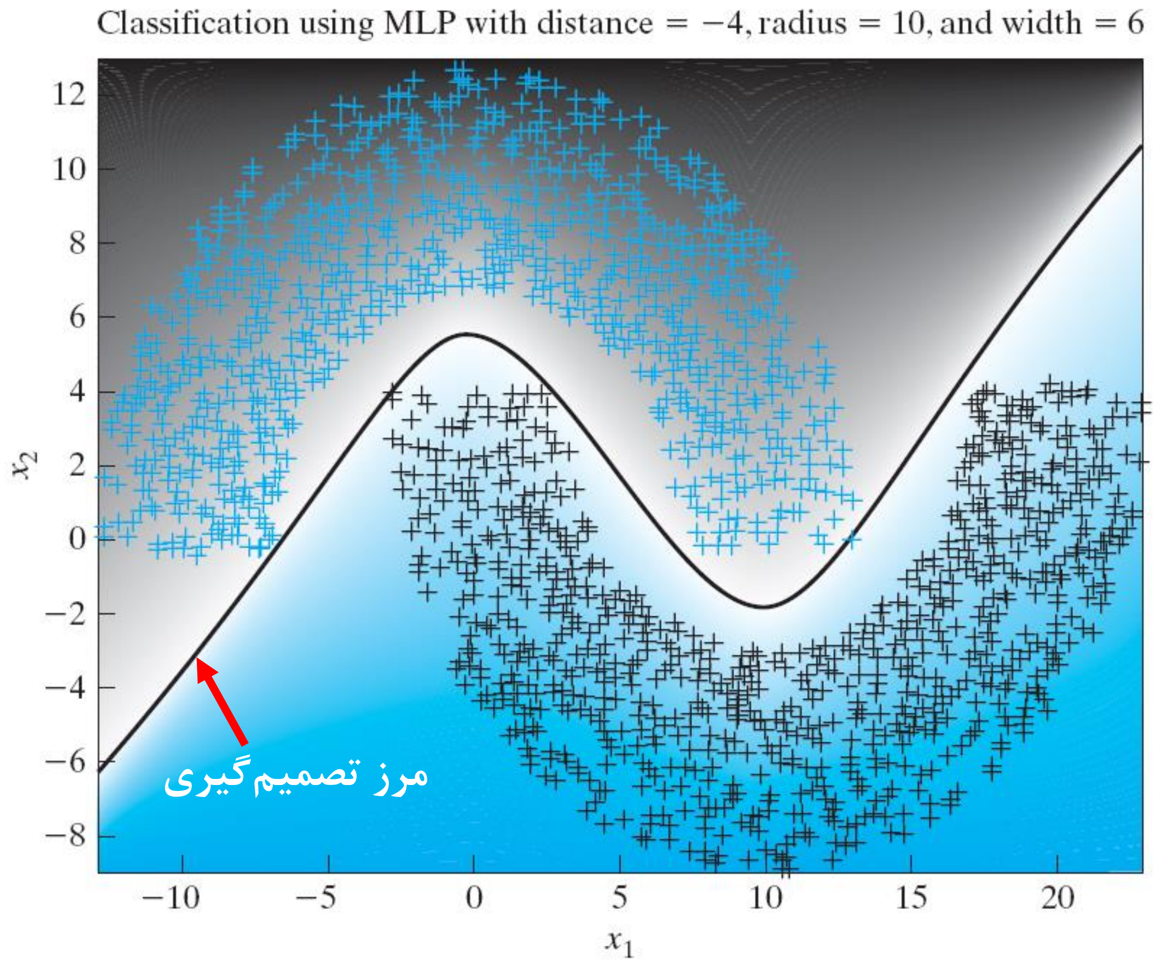
آ– چنانچه $E\left(\mathbf{w}(n) + \Delta\mathbf{w}(n)\right) \geq E\left(\mathbf{w}(n-1)\right)$ ، λ ده برابر شود. بازگشت به گام ۳.

ب– چنانچه $E\left(\mathbf{w}(n) + \Delta\mathbf{w}(n)\right) < E\left(\mathbf{w}(n-1)\right)$ ، λ یک دهم شود. بازگشت به گام ۳.

– این الگوریتم را می‌توان به روش دسته‌ای (Batch Mode) نیز اجرا کرد.

پرسپترون چندلایه (MLP)

مثال کلاسه بندی:



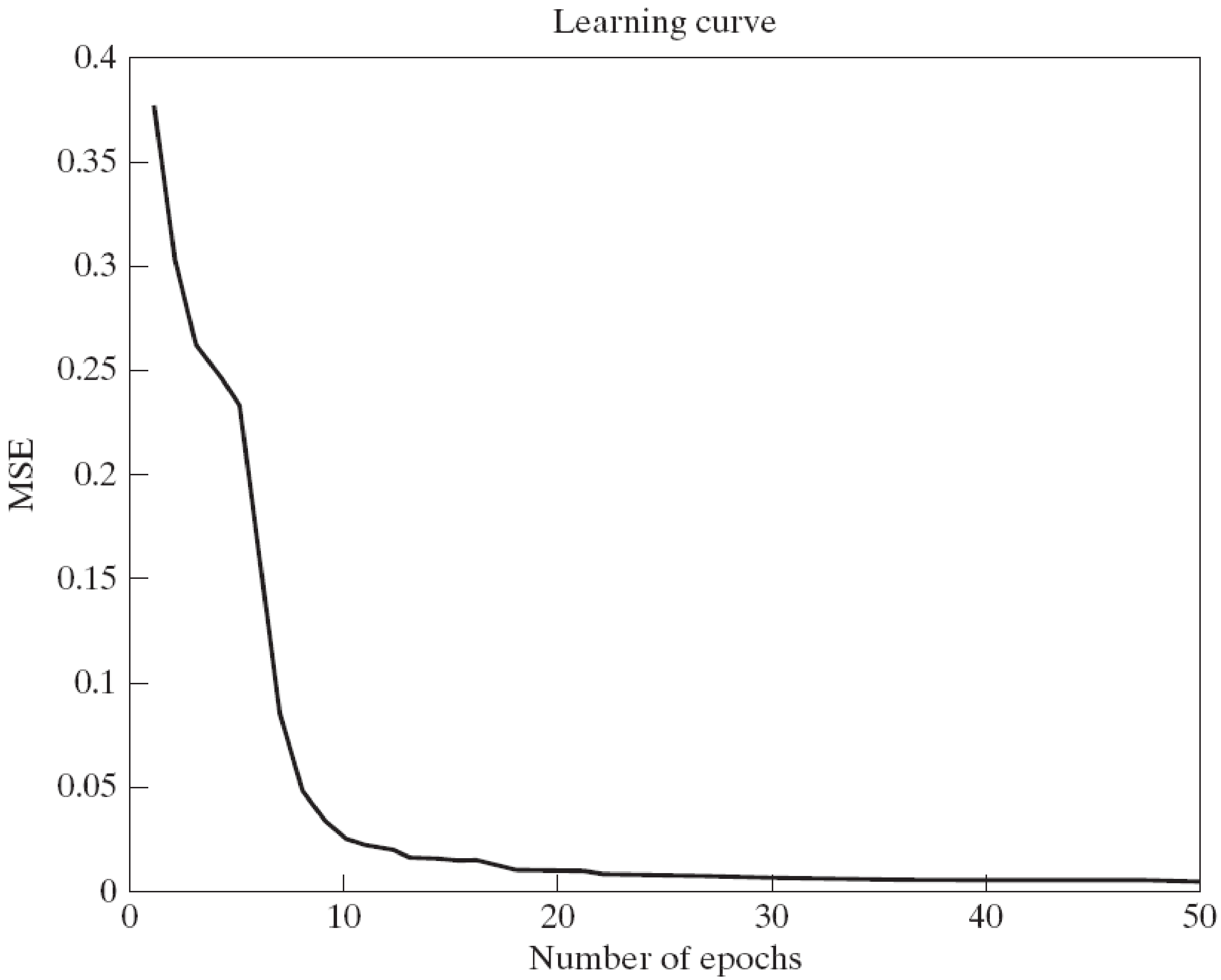
- تعداد ورودی ها: ۲
- تعداد سلول‌های یک لایه پنهان: ۲۰
- تعداد خروجی: ۱
- تابع غیرخطی سلول‌ها

$$\varphi(v) = \frac{1 - \exp(-2v)}{1 + \exp(-2v)}$$

– ضریب آموزش (η) از ۰/۱ تا $۱۰^{-۵}$ کاهش می‌یابد.

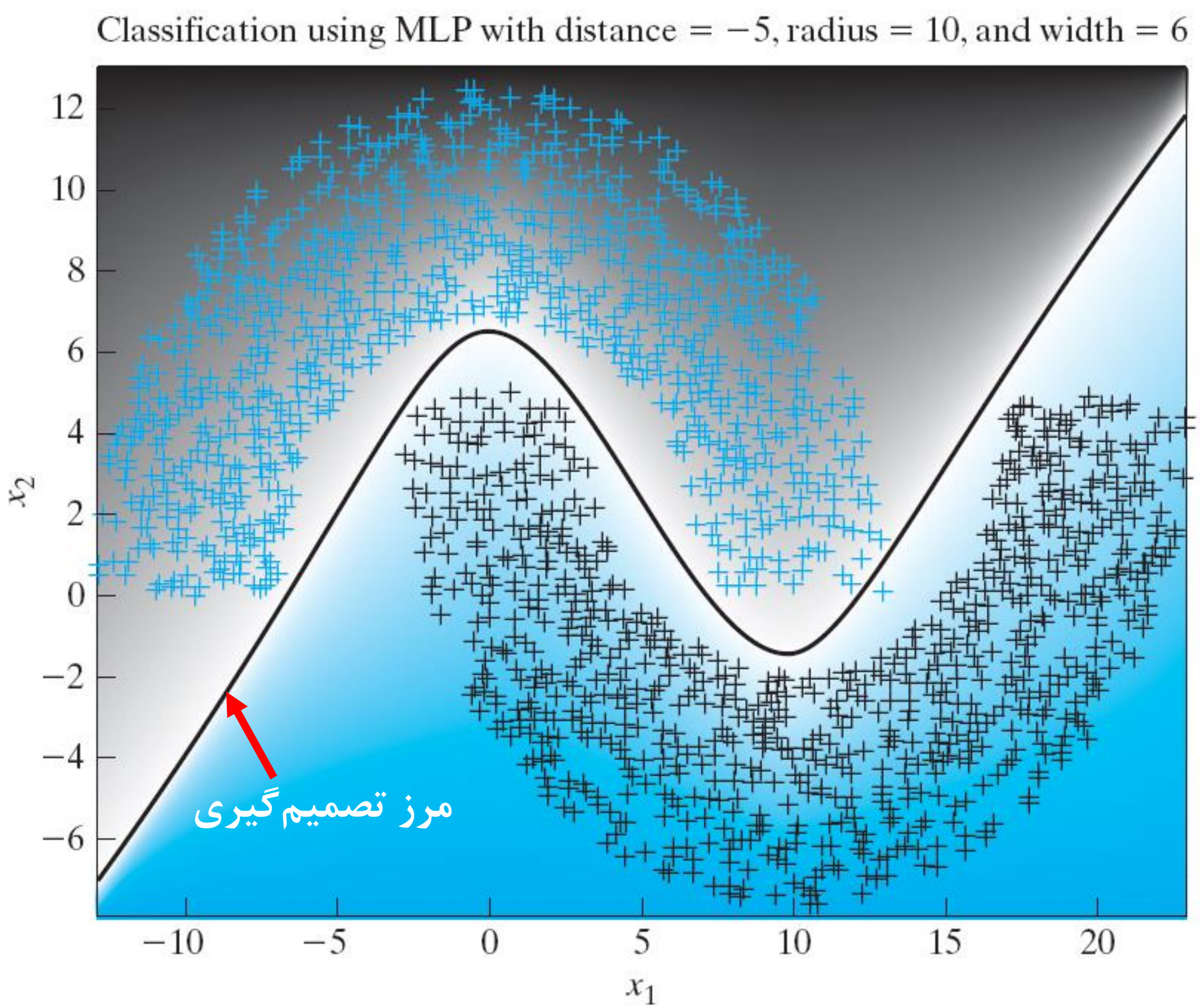
پرسپترون چندلایه (MLP)

مثال کلاسه بندی:



پرسپترون چندلایه (MLP)

مثال کلاسه بندی:



پرسپترون چندلایه (MLP)

مثال کلاسه بندی: