COURSE 02457

Signal Processing in Non-linear Systems:

Lecture 5

- Generalized backprop
- Example: NetTalk
- Local quadratic approximation to the cost function
- Weight decay and pruning
- Test error estimation and penalties
- Example: Sunspot predictions
- Gradient descent and line searches
- Conjugate gradients
- Newton's method
- The Hessian matrix and approximations

The general Backprop rule

- Consider a hidden unit $z_j = g(a_j)$, where $a_j = \sum_i w_{j,i} z_i$
- then the derivative can be expressed

$$\begin{aligned} \frac{\partial E}{\partial w_{ji}} &= \sum_{j'} \frac{\partial E}{\partial a_{j'}} \frac{\partial a_{j'}}{\partial w_{ji}} \\ &= \frac{\partial E}{\partial a_j} \frac{\partial a_j}{\partial w_{ji}} \end{aligned}$$

- Let $\delta_j = \frac{\partial E}{\partial a_j}$, note also $\frac{\partial a_j}{\partial w_{ji}} = z_i$, this leads to

$$\frac{\partial E}{\partial w_{ji}} = \sum_n \delta_j^n z_i(\mathbf{x}^n)$$

- Computing the $\delta' s$

$$\begin{aligned} \delta_j^n &\equiv \frac{\partial E^n}{\partial a_j} = \sum_k \frac{\partial E^n}{\partial a_k} \frac{\partial a_k}{\partial a_j} \\ \delta_j^n &= g'(a_j) \sum_k w_{k,j} \delta_k^n \end{aligned}$$

Local quadratic approximation

- Second order Taylor expansion of the costfunction

$$
\begin{aligned}
E(\mathbf{w}) \;\approx\; & E(\mathbf{w}_0) + \sum_j \frac{\partial E}{\partial w_j}\left(w_j - w_{0,j}\right) \\
& + \frac{1}{2}\sum_{j,k} \frac{\partial^2 E}{\partial w_j \partial w_k}\left(w_j - w_{0,j}\right)\left(w_k - w_{0,k}\right) \\
E(\mathbf{w}) \;\approx\; & E(\mathbf{w}_0) + \frac{\partial E}{\partial \mathbf{w}}\left(\mathbf{w} - \mathbf{w}_0\right) \\
& + \frac{1}{2}\left(\mathbf{w} - \mathbf{w}_0\right)^T \frac{\partial^2 E}{\partial \mathbf{w} \partial \mathbf{w}^T}\left(\mathbf{w} - \mathbf{w}_0\right)
\end{aligned}
$$

- The matrix $\mathbf{H} = \frac{\partial^2 E}{\partial \mathbf{w} \partial \mathbf{w}^T}$ is called the $Hessian$
- The local approximation to the gradient (∇E) is given by

$$
\frac{\partial E}{\partial \mathbf{w}}(\mathbf{w}) \approx \frac{\partial E}{\partial \mathbf{w}}(\mathbf{w}_0) + \mathbf{H}(\mathbf{w}_0)\left(\mathbf{w} - \mathbf{w}_0\right)
$$

Expansion around a minimum

- At a minimum $\nabla E = 0$

$$E(\mathbf{w}) \approx E(\mathbf{w}^*) + \frac{1}{2}(\mathbf{w} - \mathbf{w}^*)^T H(\mathbf{w}^*)(\mathbf{w} - \mathbf{w}^*)$$

- The Hessian is real and symmetric, hence a has a set of orthonormal eigenvectors

$$\mathbf{H}\mathbf{u}_j = \lambda_j \mathbf{u}_j$$
$$\mathbf{u}_i^\top \mathbf{u}_j = \delta_{ij}$$

- At a minimum the Hessian is positive: $\mathbf{v}^\top \mathbf{H}\mathbf{v} > 0$
- in particular for all eigenvectors

$$\mathbf{u}_j^\top \mathbf{H}\mathbf{u}_j = \lambda_j > 0$$

- hence, when the Hessian is positive all eigenvalues are positive

The generalization error

- Let a training set be given by $\mathcal{D} = \{(t^1, \mathbf{x}^1), ..., (t^N, \mathbf{x}^N)\}$.
- The mean square error of the model $y(\mathbf{x}; \mathbf{w})$ is given by

$$E = \frac{1}{2} \sum_{n=1}^{N} (y(\mathbf{x}^n; \mathbf{w}) - t^n)^2$$

- Now consider the limit of large sets, the error per example is

$$\begin{aligned} E &= \lim_{N \to \infty} \frac{1}{2N} \sum_{n=1}^{N} (y(\mathbf{x}^n; \mathbf{w}) - t^n)^2 \\ &= \frac{1}{2} \int \int (y(\mathbf{x}; \mathbf{w}) - t)^2 p(t, \mathbf{x}) dt d\mathbf{x} \end{aligned}$$

- This is the average (or expected) error on a test datum $(\mathbf{x}, t)$, which we call the generalization error.

Crossvalidation

- Generalization errors can not be measured, but can be estimated using a finite $test\ set$
- We would like to use as many examples as possible for training.
- Crossvalidation: Split the data set in V subsets $\mathcal{D}_v$

$$\mathcal{D} = \mathsf{U}_{v=1}^{V} \mathcal{D}_v \tag{1}$$

- For $v = 1, ..., V$ train on $\mathcal{D}/\mathcal{D}_v$ and estimate the test error by

$$\begin{aligned} E_{\text{test}} &= \frac{1}{V} \sum_{v=1}^{V} E_v \\ E_v &= \frac{1}{2} \sum_{n \in \mathcal{D}_v} (y(\mathbf{x}^n; \mathbf{w}) - t^n)^2 \end{aligned}$$

- Extreme (and likely optimal): $V = N$, aka leave-one-out crossvalidation

Model capacity and test errors

- The Generalization error depends on the interplay between model flexibility and training set size
- The learning curve is the relation between generalization and training set size: $E_{\text{test}}(N)$ vs. N.
- The generalization error is determined by the complexity of the model and the amount of data N.
- The model complexity is controlled by regularization and by pruning

Regularization by weight decay

- Weight decay is a means of soft capacity control

$$\tilde{E}(\mathbf{w}) = E(\mathbf{w}) + \frac{1}{2}\nu \mathbf{w}^T \mathbf{w}$$

- Analysis of weight decay: Second order Taylor expansion of the costfunction

$$\begin{aligned} E(\mathbf{w}) &\approx E(\mathbf{w}_0) + \sum_j \frac{\partial E}{\partial w_j}\left(w_j - w_{0,j}\right) \\ &\quad + \frac{1}{2}\sum_{j,k} \frac{\partial^2 E}{\partial w_j \partial w_k}\left(w_j - w_{0,j}\right)\left(w_k - w_{0,k}\right) \\ E(\mathbf{w}) &\approx E(\mathbf{w}_0) + \frac{\partial E}{\partial \mathbf{w}}\left(\mathbf{w} - \mathbf{w}_0\right) \\ &\quad + \frac{1}{2}\left(\mathbf{w} - \mathbf{w}_0\right)^T \frac{\partial^2 E}{\partial \mathbf{w} \partial \mathbf{w}^T}\left(\mathbf{w} - \mathbf{w}_0\right) \end{aligned}$$

- The matrix $\mathbf{H} = \frac{\partial^2 E}{\partial \mathbf{w} \partial \mathbf{w}^T}$ is called the $Hessian$

Weight decay

- Analysis of weight decay:

$$E(\mathbf{w}) \approx E(\mathbf{w}_0) + \frac{\partial E}{\partial \mathbf{w}}(\mathbf{w} - \mathbf{w}_0) + \frac{1}{2}(\mathbf{w} - \mathbf{w}_0)^T \mathbf{H}(\mathbf{w} - \mathbf{w}_0)$$

- Hence the minimum solves

$$\frac{\partial E}{\partial \mathbf{w}} + \mathbf{H}(\mathbf{w} - \mathbf{w}_0) = 0$$

$$\mathbf{w}^* = \mathbf{w}_0 - \mathbf{H}^{-1}\frac{\partial E}{\partial \mathbf{w}}$$

Weight decay

- Now if there is a non-zero weight decay

$$\frac{\partial \tilde{E}(\mathbf{w})}{\partial \mathbf{w}} = \frac{\partial E}{\partial \mathbf{w}} + \nu \mathbf{w}$$

- Hence the new minimum solves

$$\frac{\partial E}{\partial \mathbf{w}_0} + \mathbf{H}_0(\mathbf{w}^* - \mathbf{w}_0) + \nu(\mathbf{w}^* - \mathbf{w}_0) = 0$$

- while the old minimum solves

$$\frac{\partial E}{\partial \mathbf{w}_0} + \mathbf{H}_0(\mathbf{w} - \mathbf{w}_0) = 0$$

- this means that the new and the old minima are related as

$$\mathbf{w}^* - \mathbf{w}_0 = (\mathbf{H}_0 + \nu \mathbf{1})^{-1} \mathbf{H}_0(\mathbf{w} - \mathbf{w}_0)$$

Saliency: Optimal Brain Damage

- How much does the training error increase if we delete a weight
- Second order expansion:

$$E(\mathbf{w}) \approx E(\mathbf{w}^*) + \frac{\partial E}{\partial \mathbf{w}}(\mathbf{w} - \mathbf{w}^*) + \frac{1}{2}(\mathbf{w} - \mathbf{w}^*)^T \mathbf{H} (\mathbf{w} - \mathbf{w}^*)$$

- Deletion of the j'th weight: $\mathbf{w} - \mathbf{w}^* = w_j \mathbf{e}_j$

$$E(\mathbf{w}) \approx E(\mathbf{w}^*) + \frac{\partial E}{\partial \mathbf{w}} w_j \mathbf{e}_j + \frac{1}{2} w_j \mathbf{e}_j^T \mathbf{H} w_j \mathbf{e}_j$$

$$E(\mathbf{w}) \approx E(\mathbf{w}^*) + \frac{\partial E}{\partial w_j} w_j + \frac{1}{2} \mathbf{H}_{j,j} w_j^2$$

Saliency: Optimal Brain Damage

- However, in the minimum the first derivative is zero, hence

$$\Delta E(\mathbf{w})_{\mathrm{obd}} \approx \frac{1}{2}\mathbf{H}_{jj}w_j^2$$

  defining the OBD $saliency$

- If the retraining contribution is included (the un-pruned weights are not optimal after pruning) we get instead the OBS saliency

$$\Delta E(\mathbf{w})_{\mathrm{obs}} \approx \frac{1}{2}\frac{w_j^2}{(\mathbf{H}^{-1})_{jj}}$$

Estimating Generalization: Penalties

- The training set error is a downward biased estimate

$$E_{\text{test}} \approx E_{\text{train}} + d\sigma^2$$

- If we have used regularization (by weight decay)

$$E_{\text{test,GPE}} \approx E_{\text{train}} + d_{\text{eff}}\sigma^2$$

$$d_{\text{eff}} = \sum_{j=1}^{d} \frac{\lambda_j}{\lambda_j + \nu}$$

- The noise variance can be estimated

$$\widehat{\sigma^2} \approx \frac{2E_{\text{train}}}{N - d_{\text{eff}}}$$

- Combining we find

$$E_{\text{test,GPE}} \approx \frac{N + d_{\text{eff}}}{N - d_{\text{eff}}} E_{\text{train}}$$

Gradient descent optimization

- Objective: to solve the equation $\nabla E = 0$

$$\begin{aligned} \mathbf{w}^{(\tau+1)} &= \mathbf{w}^{(\tau)} + \Delta\mathbf{w}^{(\tau)} \\ \Delta\mathbf{w}^{(\tau)} &= -\eta \nabla E|_{\mathbf{w}^{(\tau)}} \end{aligned}$$

- η is the learning parameter
- η can be too small: convergence very slow
- η can be too large: oscillatory behavior
- Find η by line search along the search direction $\mathbf{d}^{(\tau)} = -\nabla E|_{\mathbf{w}^{(\tau)}}$:

$$\begin{aligned} \mathbf{w}^{(\tau+1)} &= \mathbf{w}^{(\tau)} + \eta^{(\tau)}\mathbf{d}^{(\tau)} \\ E(\eta) &= E(\mathbf{w}^{(\tau)} + \eta\mathbf{d}^{(\tau)}) \end{aligned}$$

Conjugate gradient method

- Objective: to solve the equation $\nabla E = 0$

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} + \eta \mathbf{d}^{(\tau)}$$

- Let $\mathbf{g}^{(\tau)} \equiv \nabla E(\mathbf{w}^{(\tau)})$, the optimal η solves

$$\begin{aligned} \frac{\partial}{\partial \eta} E(\mathbf{w}^{(\tau)} + \eta \mathbf{d}^{(\tau)}) &= 0 \\ \nabla E(\mathbf{w}^{(\tau)} + \eta \mathbf{d}^{(\tau)})^{\top} \mathbf{d}^{(\tau)} &= 0 \\ \mathbf{g}^{(\tau+1)\top} \mathbf{d}^{(\tau)} &= 0 \end{aligned}$$

- Gradient at optimal point is orthogonal to the search direction!

$$\mathbf{g}(\mathbf{w}^{(\tau+1)})^{\top} \mathbf{d}^{(\tau)} = 0$$

- Choose the search direction so that this property also holds between new and old search direction:

$$\mathbf{g}(\mathbf{w}^{(\tau+1)} + \eta \mathbf{d}^{(\tau+1)})^{\top} \mathbf{d}^{(\tau)} = 0$$

Conjugate gradient method cont'd

- Gradient at optimal point is orthogonal to the search direction! If we search such orthogonal directions we keep minimal interference.

$$\mathbf{g}(\mathbf{w}^{(\tau+1)})^\top \mathbf{d}^{(\tau)} = 0$$

- Choose the search direction so that this property also holds between new and old search direction:

$$\mathbf{g}(\mathbf{w}^{(\tau+1)} + \eta \mathbf{d}^{(\tau+1)})^\top \mathbf{d}^{(\tau)} = 0$$

- expand to second order:

$$\mathbf{g}(\mathbf{w}^{(\tau+1)} + \eta \mathbf{d}^{(\tau+1)}) \approx \mathbf{g}(\mathbf{w}^{(\tau+1)}) + \eta \mathbf{H} \mathbf{d}^{(\tau+1)}$$

$$\left(\mathbf{g}(\mathbf{w}^{(\tau+1)}) + \eta \mathbf{H} \mathbf{d}^{(\tau+1)}\right)^\top \mathbf{d}^{(\tau)} = 0$$

$$\mathbf{d}^{(\tau+1)\top} \mathbf{H} \mathbf{d}^{(\tau)} = 0$$

- This defines the conjugate directions.

Conjugate gradient method cont'd

- A complete set of conjugate directions can be found for a quadratic problem:

$$\mathbf{d}^{(\tau+1)} = -\nabla E^{(\tau+1)} + \beta^{(\tau)}\mathbf{d}^{(\tau)}$$

- with the three alternative definitions ($\mathbf{g} \equiv \nabla E$) (Hestenes-Stiefel, Polak-Ribiere, Fletcher-Reeves):

$$\beta^{(\tau)} = \frac{\mathbf{g}^{(\tau+1)\top}\left(\mathbf{g}^{(\tau+1)} - \mathbf{g}^{(\tau)}\right)}{\mathbf{d}^{(\tau)\top}\left(\mathbf{g}^{(\tau+1)} - \mathbf{g}^{(\tau)}\right)}$$

$$\beta^{(\tau)} = \frac{\mathbf{g}^{(\tau+1)\top}\left(\mathbf{g}^{(\tau+1)} - \mathbf{g}^{(\tau)}\right)}{\mathbf{g}^{(\tau)\top}\mathbf{g}^{(\tau)}}$$

$$\beta^{(\tau)} = \frac{\mathbf{g}^{(\tau+1)\top}\mathbf{g}^{(\tau)}}{\mathbf{g}^{(\tau)\top}\mathbf{g}^{(\tau)}}$$

- Furthermore, if we perform a perfect line search at every step the algorithm will converge in W steps for the quadratic problem. For the general costfunction nothing definitive is known, but it should work close to the minimum....

Newtons method in 1D

- Let the costfunction be approximated,

$$E(w) = E(w^*) + \frac{1}{2}H(w - w^*)^2$$

- The derivative is given by

$$\frac{\partial E}{\partial w}(w) = \frac{\partial E}{\partial w}(w^*) + H(w - w^*)$$

$$\frac{\partial E}{\partial w}(w) = H(w - w^*)$$

- This means that the distance from w to w^* is

$$w^* = w - H^{-1}\frac{\partial E}{\partial w}(w)$$

- Hence the optimal step is $\Delta w = -H^{-1}\frac{\partial E}{\partial w}(w)$

Newtons method in multiple dimensions

- At a minimum $\nabla E = 0$

$$E(\mathbf{w}) \approx E(\mathbf{w}^*) + \frac{1}{2}(\mathbf{w} - \mathbf{w}^*)^T \mathbf{H} (\mathbf{w} - \mathbf{w}^*)$$

$$\nabla E(\mathbf{w}) \approx \mathbf{H} (\mathbf{w} - \mathbf{w}^*)$$

- We find the optimal multivariate step is given by

$$\mathbf{w}^* = \mathbf{w} - \mathbf{H}^{-1} \nabla E(\mathbf{w})$$

- this is the Newton direction, for a quadratic problem this solves the optimization problem in one iteration!

Hessian for a least squares problem

- The least squares costfunction

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}(y^n - d^n)^2$$

- The first derivative is

$$\frac{\partial E}{\partial \mathbf{w}} = \sum_{n=1}^{N}(y^n - d^n)\frac{\partial y^n}{\partial \mathbf{w}}$$

- The second derivative is

$$\frac{\partial^2 E}{\partial \mathbf{w}\partial \mathbf{w}^\top} = \sum_{n=1}^{N}\frac{\partial y^n}{\partial \mathbf{w}}\frac{\partial y^n}{\partial \mathbf{w}}^\top + \sum_{n=1}^{N}(y^n - d^n)\frac{\partial^2 y^n}{\partial \mathbf{w}\partial \mathbf{w}^\top}$$

- The Gauss-Newton or outer product approximation is

$$\frac{\partial^2 E}{\partial \mathbf{w}\partial \mathbf{w}^\top} \approx \sum_{n=1}^{N}\frac{\partial y^n}{\partial \mathbf{w}}\frac{\partial y^n}{\partial \mathbf{w}}^\top$$

- The pseudo-Gauss-Newton approximation is to ignore the off-diagonal terms