

Neural Computation : Important Equations

John A. Bullinaria - 2011

To study Neural Networks effectively, one has to deal with quite a lot of mathematics. The examination will not require you to carry out complex mathematical derivations, nor remember complex equations, but it may require you to understand and explain some of the following important equations from the lectures:

Basic neuron equation

$$out_j = f\left(\sum_{i=1}^n w_{ji} in_i - \theta_j\right)$$

Sigmoid/Logistic activation function

$$\text{Sigmoid}(x) = \frac{1}{1 + e^{-x}}$$

Sum Squared Error and Cross Entropy cost functions

$$E_{SSE}(\{w_{kl}\}) = \frac{1}{2} \sum_p \sum_j (target_j^p - out_j(in_i^p))^2$$

$$E_{CE}(\{w_{ikl}\}) = - \sum_p \sum_j \left[target_j^p \cdot \log(out_j(in_i^p)) + (1 - target_j^p) \cdot \log(1 - out_j(in_i^p)) \right]$$

Gradient Descent – weight update equation

$$\Delta w_{kl} = -\eta \frac{\partial E(\{w_{ij}\})}{\partial w_{kl}}$$

Gradient Descent for Single Layer Perceptron

$$\Delta w_{kl} = \eta \sum_p (target_l - out_l) \cdot in_k$$

Back-propagation with Momentum

$$delta_k^{(N)} = (target_k - out_k^{(N)})$$

$$delta_k^{(n)} = \left(\sum_k delta_k^{(n+1)} \cdot w_{ik}^{(n+1)} \right) \cdot f' \left(\sum_j out_j^{(n-1)} w_{jk}^{(n)} \right)$$

$$\Delta w_{kl}^{(n)}(t) = \eta \sum_p delta_l^{(n)}(t) \cdot out_k^{(n-1)}(t) + \alpha \cdot \Delta w_{kl}^{(n)}(t-1)$$

Regularization and Weight decay/sharing

$$E_{Reg} = E_{SSE/CE} + \lambda\Omega$$

$$\Omega_{WeightDecay} = \frac{1}{2} \sum_{j,i,m} (w_{ji}^{(m)})^2$$

$$\Omega_{SoftWeightSharing} = - \sum_{k,l,m} \ln \left(\sum_{j=1}^M \frac{\alpha_j}{\sqrt{2\pi\sigma_j^2}} \exp \left\{ -\frac{(w_{kl}^{(m)} - \mu_j)^2}{2\sigma_j^2} \right\} \right)$$

Bias and Variance Decomposition

$$\begin{aligned} & \mathcal{E}_D \left[\left(\mathcal{E}[y | x_i] - net(x_i, W, D) \right)^2 \right] \\ &= \left(\mathcal{E}_D[net(x_i, W, D)] - \mathcal{E}[y | x_i] \right)^2 + \mathcal{E}_D \left[\left(net(x_i, W, D) - \mathcal{E}_D[net(x_i, W, D)] \right)^2 \right] \\ &= \quad \quad \quad (\text{bias})^2 \quad \quad \quad + \quad \quad \quad (\text{variance}) \end{aligned}$$

Radial Basis Function Networks

$$y_k(\mathbf{x}) = \sum_{j=0}^M w_{kj} \phi_j(\mathbf{x})$$

$$\phi_j(\mathbf{x}) = \exp \left(-\frac{\|\mathbf{x} - \boldsymbol{\mu}_j\|^2}{2\sigma_j^2} \right)$$

Kohonen Networks – discriminant function and weight update equation

$$d_j(\mathbf{x}) = \sum_{i=1}^D (x_i - w_{ji})^2$$

$$T_{j,I(\mathbf{x})}(t) = \exp(-S_{j,I(\mathbf{x})}^2 / 2\sigma^2(t)) \quad \sigma(t) = \sigma_0 \exp(-t/\tau_\sigma)$$

$$\Delta w_{ji} = \eta(t) \cdot T_{j,I(\mathbf{x})}(t) \cdot (x_i - w_{ji}) \quad \eta(t) = \eta_0 \exp(-t/\tau_\eta)$$

Mixtures of Experts – gated combination

$$y(p) = \sum_{i=1}^K g_i(\mathbf{x}(p)) y_i(p)$$

$$g_i(p) = \exp \left(\sum_j a_{ij} x_j(p) \right) / \sum_{l=1}^K \exp \left(\sum_j a_{lj} x_j(p) \right)$$