Deep Learning
Regularization
Weight decay

Ali Ghodsi

University of Waterloo

Regularization

A central problem in machine learning is how to make an algorithm that will perform well not just on the training data, but also on new inputs.

Most machine learning tasks are estimation of a function $\hat{f}(x)$ parametrized by a vector of parameters θ.

Classical Regularization: Parameter Norm Penalty

Most classical regularization approaches are based on limiting the capacity of models, by adding a parameter norm penalty $\Omega(\theta)$ to the objective function J.

$$J(\theta; X, y) = J(\theta; X, y) + \alpha\Omega(\theta)$$

L^2 Parameter Regularization, weight decay

L^2 parameter norm penalty commonly known as *weight decay*.

Regularization term $\Omega(\theta) = \frac{1}{2}\|w\|_2^2$

Gradient of the total objective function:

$$\nabla_w \tilde{J}(w; X, y) = \alpha w + \nabla_w J(w; X, y).$$
$$w := w - \epsilon(\alpha w + \nabla w J(w; X, y)).$$

Considering a quadratic approximation to the objective function

$$\hat{J}(\theta) = J(w^*) + \frac{1}{2}(w - w^*)^T H(w - w^*)$$
$$\nabla_w \hat{J}(w) = H(w - w^*).$$

$$\alpha w + H(w - w^*) = 0$$

$$(H + \alpha I)w = Hw^*$$

$$\tilde{w} = (H + \alpha I)^{-1} H w^*.$$

what happens as α grows?

$H = Q\Lambda Q^T$

$$\tilde{w} = (Q\Lambda Q^T \alpha I)^{-1} Q\Lambda Q^T w^*$$

$$= [Q(\Lambda + \alpha I)Q^T]^{-1} Q\Lambda Q^T w^*$$

$$= Q(\Lambda + \alpha I)^{-1} \Lambda Q^T w^*,$$

$$Q^T \tilde{w} = (\Lambda + \alpha I)^{-1} \Lambda Q^T w^*.$$

The effect of weight decay is to rescale the coefficients of eigenvectors. The ith component is rescaled by a factor of $\frac{\lambda_i}{\lambda_i + \alpha}$.

- If $\lambda_i \gg \alpha$, the effect of regularization is relatively small.
- Components with $\lambda_i \ll \alpha$ will be shrunk to have nearly zero magnitude.

Directions along which the parameters contribute significantly to reducing the objective function are preserved a small eigenvalue of the Hessian tell us that movement in this direction will not significantly increase the gradient **effective number of parameters**, defined to be

$$\gamma = \sum_i \frac{\lambda_i}{\lambda_i + \alpha}.$$

As α is increased, the effective number of parameters decreases.