Generalization and overfitting (bias/variance dilemma)

Training data
- High "variance" model
- High "bias" model
- Balanced model

Learning as Bayesian inference (maximum likelihood estimation)

\[
p(M|D) = \frac{p(D|M) p(M)}{p(D)} \quad (M = \text{model}, \ D = \text{data})
\]

\[
- \log p(M|D) = -\log p(D|M) - \log p(M) + \log p(D)
\]

Objective function = “error” + “complexity” (cost) [Ignore likelihood of data]

Weight decay

- Penalize large weights: \( \text{Cost} (w_{ij}) = \frac{1}{2} w_{ij}^2 \)

\[
\Delta w_{ij}^t = -\epsilon \frac{\partial E}{\partial w_{ij}} + \alpha \left( \Delta w_{ij}^{t-1} \right) - \lambda w_{ij}
\]

- Smaller weights \( \Rightarrow \) smaller net inputs \( \Rightarrow \) keeps units in the linear range of the sigmoid (preserves similarity)

- Similar effect caused by adding noise to activations

\[
n_{ij} = \sum_i (a_i + N(0, \sigma)) w_{ij}
\]

- Effect of noise is amplified by magnitude of weight
- Zero-mean noise on lots of smaller weights tends to cancel out

Weight elimination

- Minimize the number of connections in a network

\[
\text{Cost} (w_{ij}) = \lambda \frac{w_{ij}^2}{1 + w_{ij}^2 / w_0^2}
\]

- Tries to identify two types of weights:
  - Necessary weights: Pressure from error dominates weight decay; can be any size because decay plateaus
  - Unnecessary weights: Pressure from error is insufficient to counteract weight decay; quadratic decay back toward zero (and can be removed)

- Analogy: reaching escape velocity (defined by inflection point \( w_0 \))

Generalization and number of hidden units

- Number of hidden units constrains “flexibility” of internal representations
  - Fewer hidden units: more constrained (harder to train, better generalization)
  - More hidden units: less constrained (easier to train, poorer generalization)

- Larger representational space makes it easier to “throw away” input similarities (make more orthogonal) in converting to output similarities

- Generalization determined by degree of constraint, not by number of hidden units per se
  - Network with many hidden units but weight decay generalizes better than network with few hidden units and no weight decay
    - units can stay in linear range of sigmoid (preserves similarity)
Cross-validation ("early stopping")

- Divide training data into two sets:
  - **Training set**: Used to calculate error derivatives and change weights
  - **Validation set**: Used to determine when generalization has peaked

- Test true generalization on actual withheld **Testing set**

- Can run multiple times with different training/validation divisions and combine results