Learning representations by back-propagating errors

Key Takeaways:

1. Back Propagation was introduced as a new learning procedure through this Paper.
2. Back-propagation – Adjusts connection weights to minimize the difference between the actual and desired network output.
3. Family Tree Challenge: Network was taught relations from two isomorphic family trees (English and Italian). Information was given as triples (Person 1, Relationship, Person 2). It was trained on 100 of the 104 possible triples.
4. Hidden Units that aren't part of the input or Output, but they represent important task features below that are not at all explicit in the input:
    a. Unit 1 – English vs Italian (Nationality)
    b. Unit 2 – Generation
    c. Unit 6 – Branch of Family
5. How it works (Learning Process):
    a. Forward Pass: Input flows through the network to determine the Output.
    b. Calculate Error: The Network's output is compared with desired output.
    c. Backward Pass: Error derivatives are propagated backward through the network (top layer back to the bottom one)
    d. Adjust Weights: Connection weights are changed to reduce the error.
6. Gradient Descent: Goal is to find the set of weights that results in the lowest possible total error.
7. Power of Back-propagation:
    a. Systematically minimized errors between actual and desired outputs.
    b. Allows hidden units to represent abstract features of a task
    c. Constructs appropriate internal representations for any domain.
8. The most obvious drawback of the learning procedure is that error-surface may contain local minima so that gradient descent is not guaranteed to find a global minimum.
9. The paper explains that the back propagation learning procedure for layered networks can be mapped to a learning procedure for recurrent networks by treating each time-step as a layer.
10. Task Examples: The paper demonstrates the back-propagation procedure on two tasks:
    a. Detecting Symmetry: A network with an intermediate layer learns to detect symmetry in a one-dimensional array of binary input units, a task that can't be done by a network without hidden units.
    b. Family Trees: A network is trained to store information from two isomorphic family trees. The network learns to encode people and relationships as distributed patterns of activity in the hidden layers, which allows it to generalize correctly to untrained cases by making use of the family tree isomorphism.

Tools: NotebookLM helped me review and understand the details of this paper