04 / 26 / 2025

1. Deep Learning

i)

=> Batch normalization addresses the problem of internal covariate shift; which slows down the training process and forces use of smaller learning rates and careful weight initialization.

Batch normalization accelerates training by reducing internal covariance shift. By normalizing layer inputs, BN transforms the data to have a $\mu = 0$ and $\sigma = 1$.

- Reducing the shift allows the network to train for fewer epochs to achieve convergence
- Reduces the need for Dropout leading to better generalization.
- BN makes network makes training scale-invariant; allowing use of larger learning rates w/o divergence.
- BN makes network less sensitive to weight initialization.

ii)

=> • Batch normalization has high dependency on mini-batch size. This is problematic in online learning tasks & large distributed models where mini-batch size must be small.

- In scenarios like RNNs, where the sequence length varies, BN isn't as effective due to lack of consistent mini-batches
- Storing separate statistics for each time step is computationally expensive

Layer Norm addresses these flaws by:

> LN does not impose constraints on mini-batch size and works for cases where batch size varies.

> LN computes statistics separately at each time step avoiding the need to store separate statistics.

iii)

=> Inductive bias refers to the set of assumptions that a model uses to generalize from a limited set of training data to unseen data.

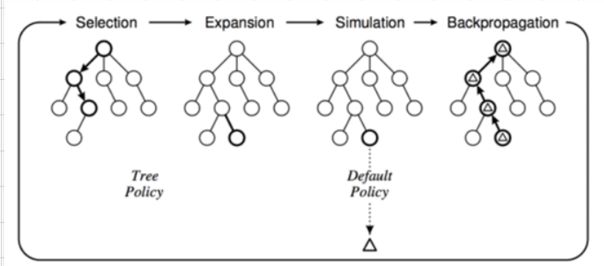
CNNs inherently have strong image specific inductive biases such as:
- Locality
- Two dimensional neighborhood structure
- Translation equivariance

As a result ViT do not generalize well when trained on insufficient amount of data and thus require larger pre-training datasets.

2. Reinforcement Learning

i>

=> Monte Carlo Tree Search (MCTS) is a heuristic search algorithm used for decision process particularly in context of games. It combines precision of tree search with power of random sampling to evaluate potential outcome of moves.
The four main steps involve:
- Selection
- Expansion
- Simulation
- Back propagation.



Alpha Go improves MCTS by integrating it with deep neural networks, specifically policy and value networks. PN, VN

- For each move, MCTS initializes a new search tree with current game state as root node.
- During selection, PN is used to prioritize moves. UCT formula is modified to add PN's move probabilities, guiding tree traversal to promising nodes
- When expanding a node, PN gives a probability distribution over possible moves, allowing AlphaGo to focus on subset of high probability moves
- Alpha Go uses Fast rollout policy to simulate game.
- During simulation, VN evaluates non-terminal states providing immediate feedback.
- Outcomes are then backpropagated through the tree, updating statistics of the nodes.

ii>

=> Policy Network is trained through supervised learning on professional game data.

After this, the policy network is improved by policy gradient RL; identical to REINFORCE (Williams 1992) with a baseline.

iii> Steps included in i>

=> The value network is used at the leaf nodes to reduce depth of the tree search.

The policy network is used to reduce the depth of the search from a node (guiding towards promising actions).