Chip Design with Deep Reinforcement Learning [2020]

Azalia Mirhoseini, Anna Goldie, Mustafa Yazgan, Joe Wenjie Jiang, Ebrahim Songhori, Shen Wang, Young-Joon Lee, Eric Johnson, Omkar Pathak, Azade Nazi, Jiwoo Pak, Andy Tong, Kavya Srinivasa, William Hang, Emre Tuncer, Quoc V. Le, James Laudon, Richard Ho, Roger Carpenter & Jeff Deana

Deep Reinforcement Learning

Deep Reinforcement Learning (DRL) is a branch of Artificial Intelligence that simulates learning through trial and error. DRL involves an AI agent navigating an environment, taking actions to maximize rewards while learning from feedback. By mimicking how humans learn from interactions, DRL enables AI to optimize actions and achieve goals effectively, making it pivotal for tasks requiring intelligent decision-making.

Floor Planning Problem

The Chip Floorplanning Problem involves arranging modules on a computer chip to optimize power, performance, and area while adhering to constraints. This task is complex due to the intricate circuitry, large-scale netlist graphs, and significant computational time needed. Despite research, human experts still spend weeks iterating for viable solutions using electronic design tools.

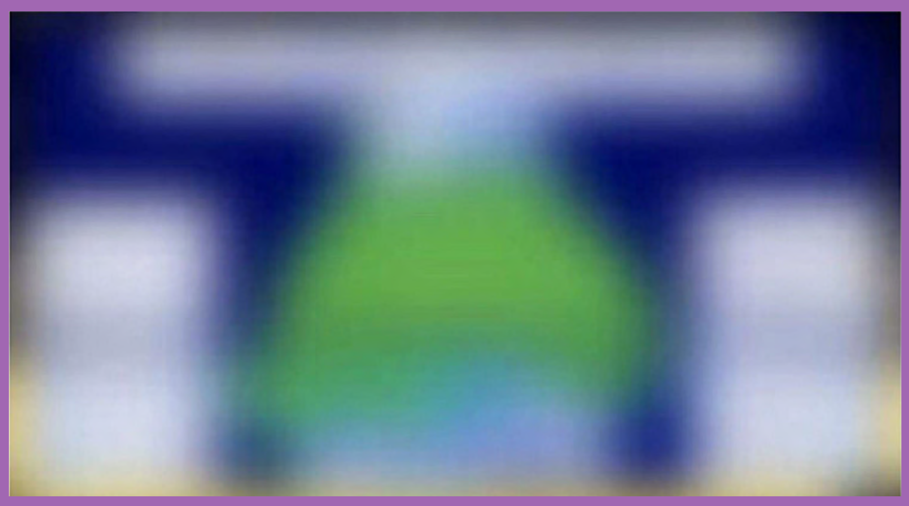
Abstract

The Deep Reinforcement Learning Model acts as a helpful guide for chip design. It processes data about the chip's components, the current piece being placed, and the specific details of the chip. Using a specialized network, it understands the chip's state and the new piece's role. This understanding aids in deciding the optimal position for the piece on the chip.

During training, the model arranges the vital components step by step. It then employs an effective technique for placing the remaining parts. This technique resembles organizing interconnected springs to optimize placement. The model refines its skills through rapid evaluations of its placements, considering parameters like wire length and congestion. This learning process enhances its chip design abilities, leading to improved efficiency and performance.

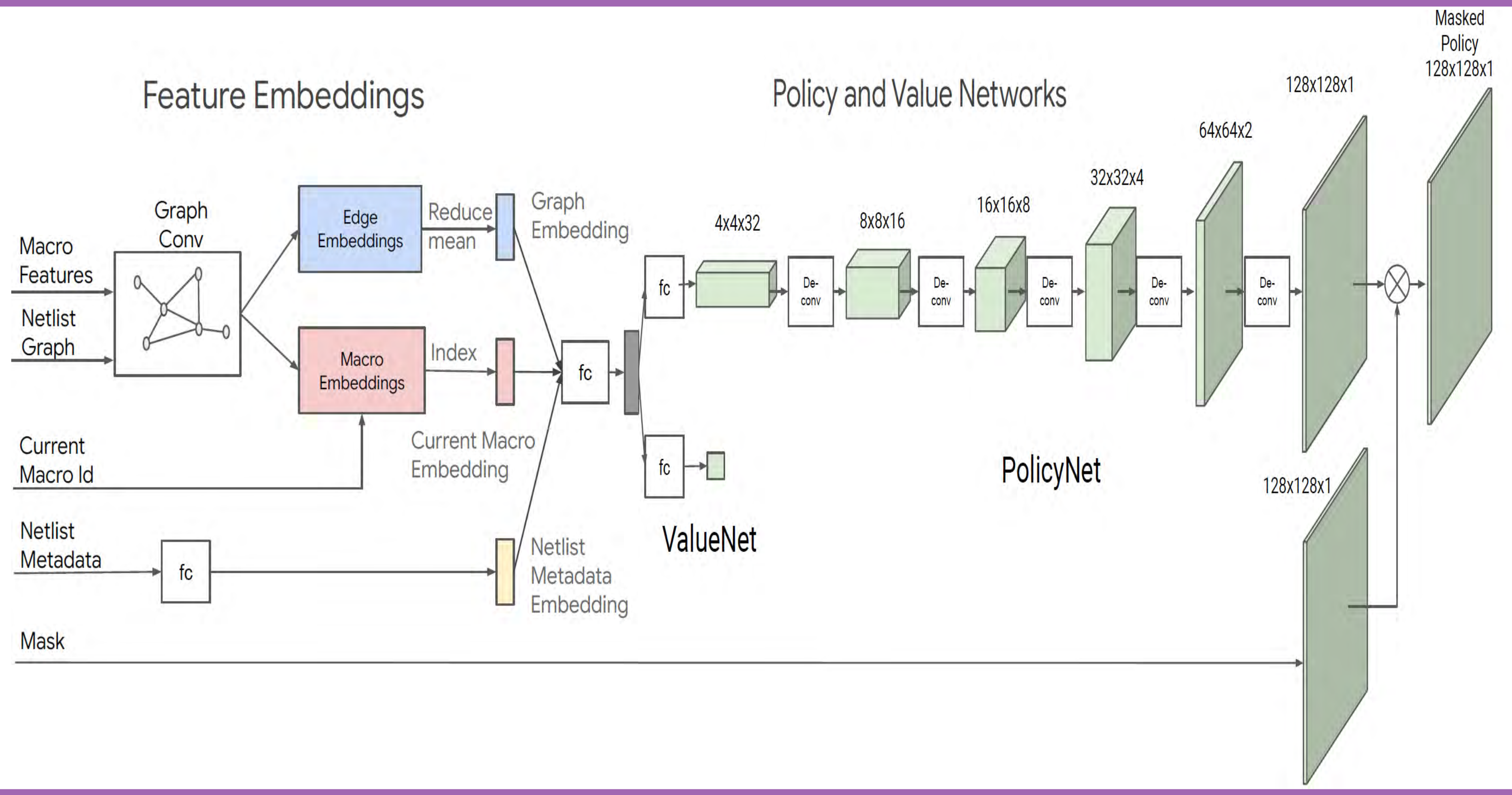
Using deep reinforcement learning transforms chip design by training an agent to intelligently place chip components, resulting in significant improvements in power, performance, and area metrics to traditional methods.

Human design



AI design

Network



Feature embeddings

Feature embeddings refer to a way of simplifying intricate parts of chip design into shorter, but still useful, versions. These shorter versions capture the most important details, helping AI agents understand and decide where to place components. By using feature embeddings, the process of solving complex problems becomes smoother, which, in the case of deep reinforcement learning techniques, leads to better learning outcomes.

Policy and value Embeddings

The policy and value network architecture encodes chip information and current placements. The policy network determines suitable positions for new parts, while the value network estimates the quality of those placements. Employing a specialized neural network, rewards for placing parts on new chips are predicted. This is achieved by utilizing an edge-based graph neural network, which enhances predictions even for unfamiliar data by effectively representing chip parts.

Results

The results of the study indicate that using pre-trained policies leads to better chip component placements compared to starting from scratch. The pre-trained policies consistently outperform scratch-trained ones, even with short additional training. Moreover, training on larger sets of chip blocks improves placement quality and training efficiency. This suggests that the approach has the potential to enhance chip design by learning from existing data and generating optimized placements.

Personal Perspective

The paper introduces a potentially revolutionary method in chip design by utilizing deep reinforcement learning and feature embeddings. While this approach holds promise for enhancing efficiency and performance, it also raises concerns about potential challenges. The reliance on learned representations could introduce unexpected errors or biases, highlighting the need for thorough validation and safeguards in implementing this method. Balancing the potential benefits with careful consideration of possible pitfalls will be crucial as this technology advances.