Teaching a Computer Tetris

Last summer, my friend and I played around with trying to implement simple neural networks to teach a computer how to play Tetris. Our amateur program could randomly clear a line about once every fifty iterations, but its progress would then seemingly disappear. It would also throw temper tantrums from time to time by simply rushing each block straight down, thus automatically losing. It did this after losing dozens of times without clearing a line.

I plan to make a more dedicated effort with this project. I will start with a simple Tetris or Tetris-clone program, of which there are many open-source versions, and remove entirely the user interaction. The game will automatically start, and the computer will be able to move blocks freely as the game repeats itself indefinitely. I will add methods that act as guidelines, that will determine how the computer should manipulate the blocks based on the current state of the board, the current block, and the next block that will fall. My hope is that when the project is complete, the computer will at least be able to clear a line after about ten to fifteen iterations. Because of the variations in every game and every board, I'm not sure how well I'll be able to teach the computer overall.

Neural networks operate in layers. The input layer feeds inputs to the hidden layer, which then returns results in the output layer. If I can make the hidden layer good enough at reading the inputs listed earlier, I am confident that the outputs should reflect a basic understanding of how the game works from how the computer reads it. Therefore, the hidden layer will represent most where I plan on focusing my attention for the project. In short, I plan on turning raw inputs from the game into sophisticated outputs.