Neural Networks Journey: From Bigrams to WaveNet

This repo is a collection of my work building and debugging neural networks from the ground up. The focus was on moving past "black box" libraries to actually understand the math and architecture behind sequence modeling.

Key Milestones

- **The "Makemore" Foundations:** I started with character-level language models. This involved building simple MLPs to predict the next character in a sequence—specifically using a dataset of Pokémon names to see how the model learns naming conventions.
- **Manual Backpropagation:** One of the most challenging parts was the week4_backprop notebook. Instead of relying on .backward(), I manually derived the gradients for the entire network. It's tedious but definitely the best way to understand how the chain rule actually flows through a model.
- **Stability with BatchNorm:** I implemented Batch Normalization to fix the "dead neuron" and initialization issues I ran into earlier. It made a massive difference in how fast the models converged and how sensitive they were to initial weights.
- **Scaling with WaveNet:** The final phase involved moving to a WaveNet-style architecture. By using a hierarchical structure (folding the context window), the model can "see" more of the previous characters without the computational cost of a massive flat layer.

Takeaways

The biggest win here wasn't just getting the code to run, but seeing the evolution of performance. Moving from a basic bigram model to a structured WaveNet made the generated Pokémon names feel much more "real" and less like random gibberish. It also highlighted just how important proper initialization and normalization are for training deeper networks.