

Building a Neural Network from Scratch

Dylan Smith

CSPB 3122 Professional Development – Spring 2024

Vision Statement

For this project, I want to build a Neural Network in Python from scratch (i.e. just using numpy and math). AI is a paradigm shift that will change many fields, but especially those of data scientists and programmers. Most of these new AI techniques (maybe all) use neural networks to learn from data, so deeply understanding NN's by building one from the ground up will allow me to understand how to build and when to use them. As of now I only have a vague understanding of how these work, so by building one I hope to understand (as much as possible) how they work as well as the different considerations that go into building one (i.e. hyper-parameters, type of data, etc.). By understanding the fundamentals of NN's I will be better able to leverage them in different positions for different use cases.

Motivation

I got into this post-baccalaureate program just before the release of ChatGPT. I thought that it would give me valuable skills I could use in just about any field. I still think that and value the different ways that learning CS has taught me to think, but more and more you are hearing about how much code is being implemented now by LLM's and how much this technology is disrupting the discipline. I've hear a lot about just how revolutionary this technology is and how much it will replace so rather than fighting it, I want to develop my skills and understanding so that I can leverage it.

Beyond that negative motivation though, I am fascinated by the idea that we are beginning to be able to create intelligence. I have always been fascinated by philosophy and the mind, so the idea of learning how to build intelligent systems is a practical way to understand more of those ideas. In addition to the potential to learn more about intelligence, being able to build something that can leverage intelligence is such a powerful idea and has so many interesting and impactful use cases that having the ability and know how to implement them gives one way more potential to do something useful and meaningful.

Learning Objectives

1. The first step in this project will be researching the terms and math that goes into a neural net. I currently have a base understanding of the algorithm, but don't really understand the math or the functionality of different hyperparameters. In this step I want to go through the different processes of NN's in order and define/research the terms so I can have a conceptual understanding of what each step I will be implementing does.
2. Once I understand the terminology, I want follow this youtube video ([Building a neural network FROM SCRATCH \(no Tensorflow/Pytorch, just numpy & math\) \(youtube.com\)](#)) to create a base NN from which to implement further functionality. This video goes through the math and programming of a NN that uses the "Digit Recognizer" dataset from Kaggle which I will also use for initial steps.
3. The next step once I have a functioning NN is to build a visualization program using NetworkX. The idea is to visualize the structure of the NN and then to label each node with a color on a gradient corresponding to the possible range of values (weights) that each node in the NN can

take. I want to implement this program in such a way that you can see how the weights change during the training of the NN as well as the final weights that can be used to compare the resulting NN's from further steps of testing different hyper-parameters.

4. The next step will be to implement different hyper-parameters (also from scratch) in order to see how each one affects the training of the model. This can be measured by training rate as well as model performance, both how performance improves throughout training as well as how well the end result performs.
5. Finally (with enough time) I want to test my NN on different input data to understand how you have to configure a NN for different use cases. Even for picture input data that where the pictures are converted to pixels before being trained on, what you want to learn from the data will be different for each use case. That step of understanding how to configure a NN for the specific data you are using is something I don't have a grasp on so through this step I hope to understand more how to actually apply this technology in different scenarios.

Risks to Project Completion

1. One of the major risks I see to completing this project is compute power. I'm not entirely sure how my personal computer will fair with the compute necessary to train something like this, especially if my implementation of it is inefficient.
2. Another risk that I see for this project is my limited understanding of CPU's vs GPU's. Training NN's in computationally expensive and everyone says that GPU's are needed for the process but I don't understand why. Even using Google Colab (mentioned in risk mitigation), I don't currently understand why that is needed or how to configure my NN to work efficiently with that. Or how to configure the Colab environment so that this process runs as well as it can.
3. Further risk inherent to a project like this is I don't have much understanding of the time that will go into any of these steps. I would love to get through all 5 of my learning objectives and it seems possible, but I'm not sure how hard this will actually be. Because of that, I'm not sure how much of this I will get to in the 45 hour allotment we have for the project.
4. Another risk I foresee is during the visualization step. I'm not entirely sure what to measure as each node is measuring a different feature (I think) and nodes in different layers might have different measurement criteria.

Risk Mitigation Strategy

- In order to address the first risk above, I will be building out my NN using Google Colab. When I worked with NN's in a past AI class, we used this as it gave you access to GPU's which can run the training process faster.
- In order to fully utilize google colabs resources, during my first learning objective, I also want to research the difference between CPUs and GPUs. How to access and configure the colab environment and what that means to be using GPU's as well as learning if there are any adjustments to my implementation I have to consider when running on one or the other.
- In order to mitigate the time risks associated with not fully grasping the time considerations for this project, I will set up my project using OOP so as to easily integrate different functionalities. I will build a structure where testing is easily implemented and results are given and saved in such a way as to be able to retrieve them and compare them easily. In step 4 where I will implement

different hyper parameters, I will at most implement one other hyper-parameter to understand how training changes while still leaving myself enough time for step 5.

- The risk in not being clear on the visualization step is something that I will have to just work through as I learn more and get to that step. It may be that the idea I have for this isn't reasonable or won't tell me what I think it will but I won't be able to address that until I reach that step.

- **Evaluation Criteria**

1. Initial NN implementation performs similarly to the one in the video.
2. Initial NN is designed in such a way as to easily integrate different hyper parameters as well as visualization program step 3 learning objective.
3. Visuals accurately follow how node weights change through color change during learning. (and save a video of the training?)
4. Different hyperparameters are implemented correctly and result in a well-trained model. Results are shown and saved in such a way as to be able to compare the training metrics across different hyper-parameters.
5. Model can be built so as to either:
 - a. Work with different data as input
 - b. Be copied and then adjusted so as to create a model tailored for a different dataset as input.

Project Portfolio

[Welcome to my website! | Dylan Smith \(33dylansmith33.github.io\)](https://33dylansmith33.github.io)