Carleton University

Capstone Project Final Report

Accelerated Design of Silicon Photonic Devices with Artificial Neural Networks

Partners:
Jonathan Levine
Jacob Ryall
Deng Mading
Yu Wan

Supervisors:
Winnie Ye
Leonard MacEachern

Author: Jonathan Levine

April 12th, 2022

Abstract

An Artificial Neural Network (ANN) was developed with Python's PyTorch machine learning framework to model the transmission spectrum of a silicon-on-insulator (SOI) grating coupler. The ANN model is able to simulate the transmission spectrum of an SOI grating coupler 1074 times faster than the traditional Finite-Difference Time-Domain technique. The ANN model was then used to find the optimum grating coupler parameters for light with a wavelength of 1.55 μm in the TE mode.

Contents

List of Figures

Acknowledgements

I would like to thank my two supervisors Professor Leonard MacEachern and Professor Winnie Ye for all their support, help and advice over the past academic year. I would also like to thank all my colleagues, Randi Wan, Deng Mading and Jacob Ryall who worked with me on this project.

1 Introduction

1.1 Overview and Background

1.1.1 Grating Couplers

The silicon-on-insulator (SOI) platform allows for ultra compact light confinement in densely integrated photonic circuits. The ultra compact nature of the SOI platform causes lossy coupling of light into and out of photonic circuits due to the modal mismatch between the optical fiber and the standard single-mode SOI waveguide. Grating couplers offer a solution to this problem as they are able to capture the entire optical mode of the light and diffract it into the SOI waveguide [1].

Figure [1] below illustrates how the grating coupler captures both modes of the green incoming light ray and diffracts them both into the directional coupler of the photonic circuit.

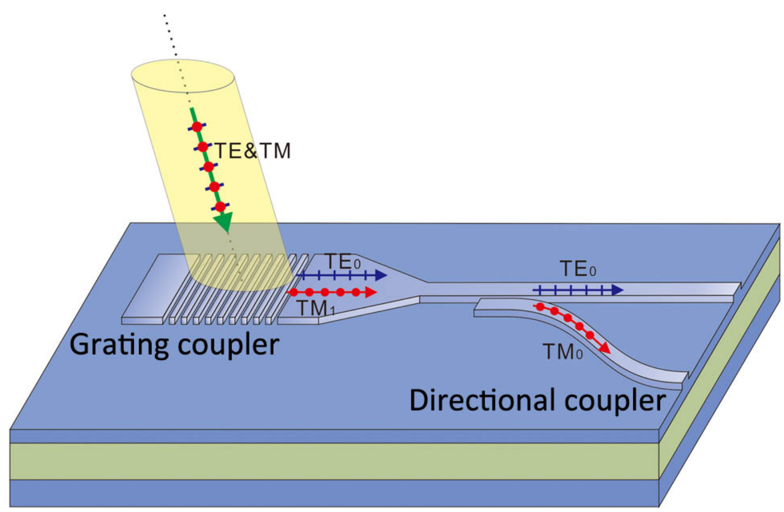

Figure 1: Silicon-On-Insulator Grating Coupler [2]

As illustrated in Figure [2], grating couplers are characterized by four physical parameters: Fiber Angle, SWG Fill Factor, Duty Cycle and the Pitch as well as the modes of the incident light beams (transverse electric TE or transverse magnetic TM) and the wavelength of the light ray.

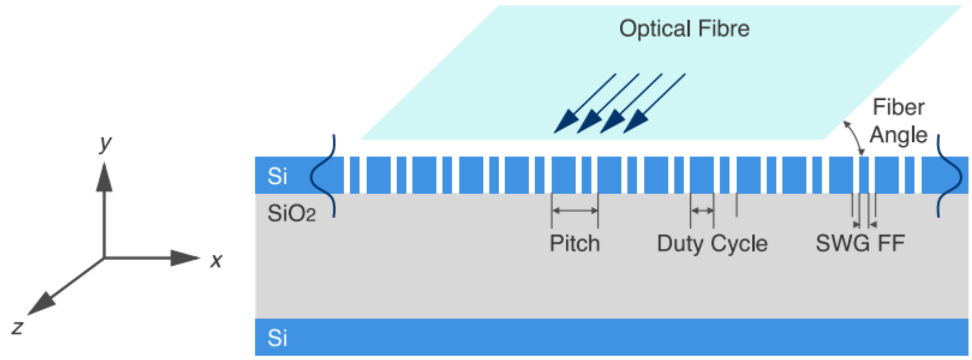

Figure 2: 2D Cross Section of an SWG Coupler [1]

An example of the kind of output spectrum we are looking for in a grading coupler is shown in Figure [3] below. We can see that this curve is a Gaussian-like bell shaped curve centered around a specific wavelength.

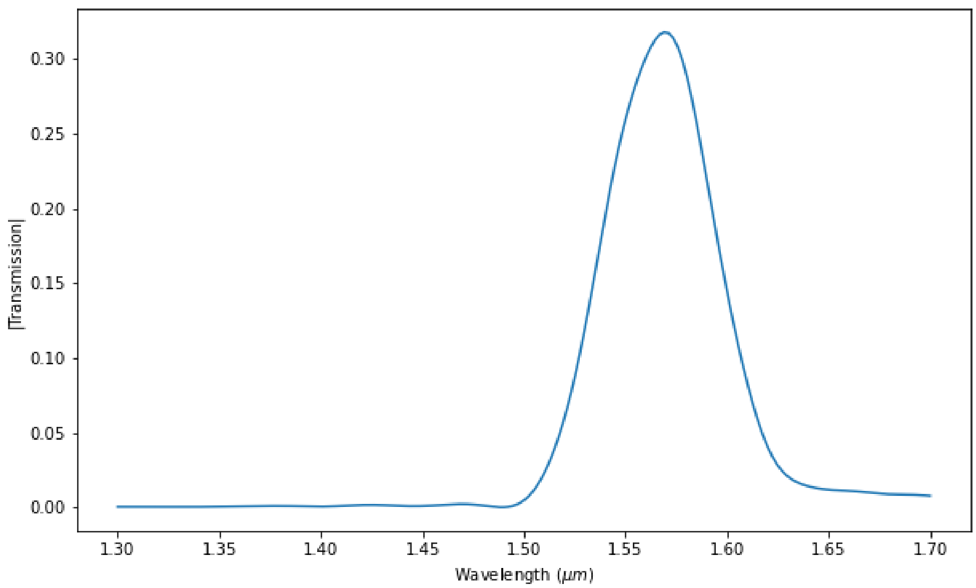

Figure 3: SWG Grating Coupler TM Mode Transmission Spectrum. The grating coupler parameters were set to: Fiber Angle = 14.375°, Pitch = 0.75μm, Duty cycle = 0.65 and Fill Factor = 0.25. [Jonathan Levine]

1.1.2 Artificial Neural Networks

Artificial Neural Networks (ANNs), are an approach to machine learning in which a computer "learns" the relationship between a system's inputs and outputs through a network of nodes, weights and activation functions. Figure[4] is a diagram of a feed-forward neural network where the orange and yellow coloured circles represent the nodes in the network, the black circles represent the bias for each layer and the grey circles represent the activation functions at each node.

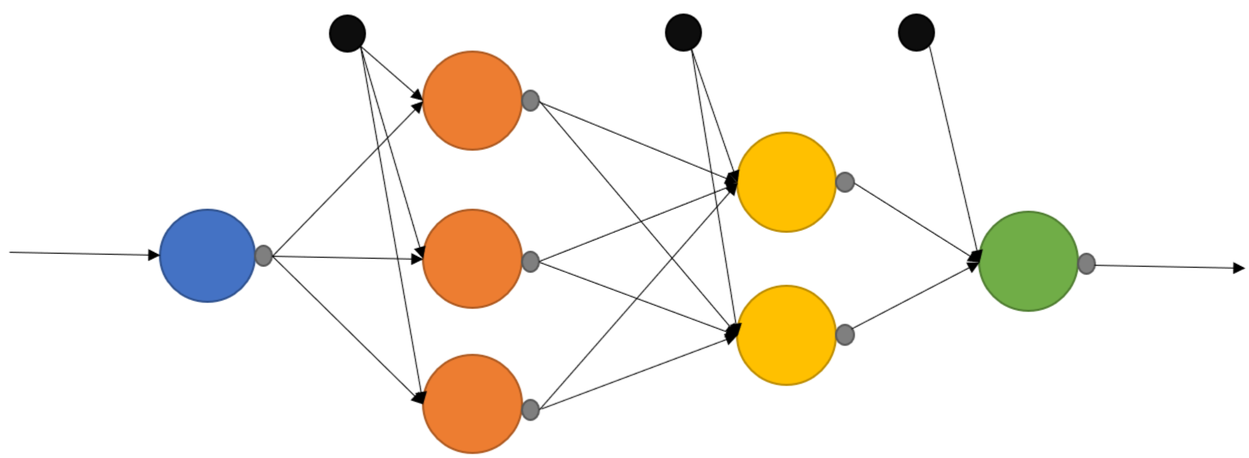

Figure 4: Feed-Forward ANN Architecture [Jonathan Levine]

Each of the nodes or *perceptrons* in the hidden layers, are connected to the next layer by weights and activation functions. These activation functions can be non-linear enabling the entire network to learn non-linear relationships between inputs. A popular non-linear activation function is the Rectified-Linear-Unit (ReLu) function shown in Figure [5] below.

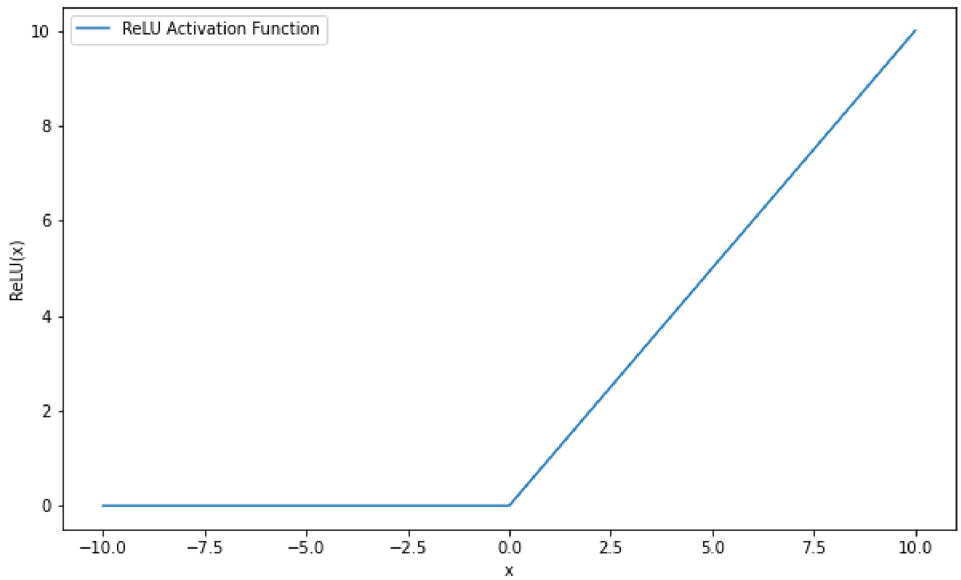

Figure 5: Rectified Linear Unit Activation Function [Jonathan Levine]

At first the weights and biases throughout a network are assigned randomly, and thus the network does not predict the outputs accurately. Mathematical algorithms are used during training to update the weights and biases in a neural network so that a loss function is minimized. These algorithms are known as **optimizers** and some popular optimizers are Stochastic-Gradient-Descent (SGD) and Adam.

1.2 Motivations and Significance

As photonic circuit requirements change, new grating couplers need to be designed to match the needs of the photonic circuits. The issue here is that performance of grating couplers is highly non-linear with respect to the wavelength, polarization (TE mode or TM mode), and incident angle of the coupled light.

The design of a grating coupler is reliant on Finite-Difference Time-Domain (FDTD) simulations. These simulations are done in software like Lumerical [3] which has a steep learning curve. These FDTD simulations are also expensive on human resources as they take days to weeks to complete. FDTD simulations require significant computational resources which drives up energy consumption. And finally Lumerical software licensing is expensive, on the order of ten thousand dollars per year.

1.3 Design Objectives and Specifications

The objective of this project is to develop an ANN model of a silicon-on-insulator grating coupler and to use this model to design a grating coupler that is optimized for incident light with a wavelength of 1.55 μm. The ANN model would provide significant time savings over the conventional FDTD numerical techniques done in Lumerical.

The ANN model would take the grating coupler parameters: fiber angle, pitch, duty cycle, fill factor, light mode and wavelength and generate an emission spectra similar to the one in Figure [3]. A simplified diagram of the ANN is shown in Figure [6] below.

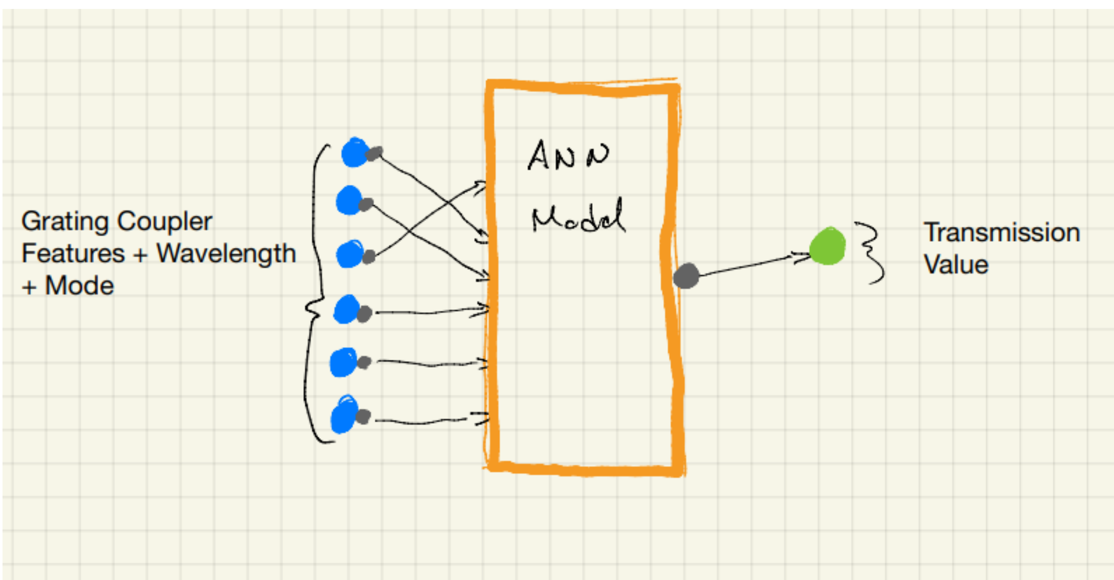

Figure 6: Hand Drawn Grating Coupler ANN Model [Jonathan Levine]

1.4 Relevant Publications

Researchers have become more and more interested in developing ANN models for Computer Aided Design. This is because ANN's can offer quick and accurate simulation performance compared to traditional numerical techniques [4].

Recently ANN's have been applied to FDTD simulations of optical devices and they have shown to offer promising results. For example the paper published by Professor Winnie Ye and a former PhD student at Carleton University Dusan Gostimirovic in their paper *An Open-Source Artificial Neural Network Model for Polarization-Insensitive Silicon-On-Insulator Grating Couplers* [1].

2 Professional Considerations

As fourth year engineering students finishing our degrees we have learned the importance and significance of the ideals and practices of Canadian Professional Engineers.

Since our programs are accredited Canadian engineering schools we will have the opportunity to get a license to practice as professional engineers. Professional Engineers Ontario (PEO) is the licensing body for professional engineers in Ontario. Over our studies we have learned the immense responsibility that comes with working as a professional engineer and the Code of Ethics that we must adhere to over our career.

In this project we applied many of the principles of the Code of Ethics of professional engineers and they are described in the next section.

2.1 Engineering Professionalism

The Code of Ethics of Canadian Professional Engineers holds Canadian Engineers to a high standard of honor and integrity and outlines the basics tenets that professional engineers must adhere to. During this project we demonstrated the following tenets of the Engineering Code of Ethics:

2.1.1 Conducted ourselves with equity, fairness, courtesy and good faith towards clients, colleagues and others

During this project we separated out tasks in an even fashion and we conducted ourselves in good faith to our supervisors and to our colleagues. We gave credit where credit is due and were always open to feedback from our supervisors.

2.2 Society and Environment

As described in the Introduction (Section [1]), the design of grating couplers currently requires intensive FDTD simulations. These simulations are driven by powerful computer hardware that draws a significant amount of power. The aim is that designing an ANN model that can replace, or diminish, a designers reliance on FDTD simulations would ultimately result in less energy consumption. With more energy efficient techniques we can contribute to the developing field of green technologies.

2.3 Project Management

At the beginning of the academic year we prepared a Gantt chart (Figure [7]) which broke down the overall timeline of the design cycle of the project. The design cycle of the ANN can be found in the Gantt chart in Figure [8] below. Every Friday afternoon from 2:30pm to 3:30pm the whole team met over Zoom to discuss our progress over the past week and discuss any issues we were having.

The Gantt chart proved to be conservative in its outlook. Overall we stuck to the projected timeline of the project and were fairly up to date with each step.

The design of the ANN for this project was broken down into two main sections: **generating simulation data** and **ANN development**. The task of generating simulation data was done by Deng Mading and Randi Wan and the ANN development was be done by Jacob Ryall and myself, Jonathan Levine. The simulation data was generated from Ansys' FDTD

Lumerical software.

Members	Individual Objectives
Yu Wan & Deng Mading	Generating Simulation Data
Jacob Ryall & Jonathan Levine	ANN Development

Table 1: Individual Objective Breakdown

For developing the ANN we had a few options to how to program it. For example we had MATLAB as an option that has a built-in machine learning toolbox. We decided on Python's PyTorch machine learning library because both Jacob and I are very familiar with programming in Python, PyTorch is also open source, and PyTorch has become one of the standard libraries in the field.

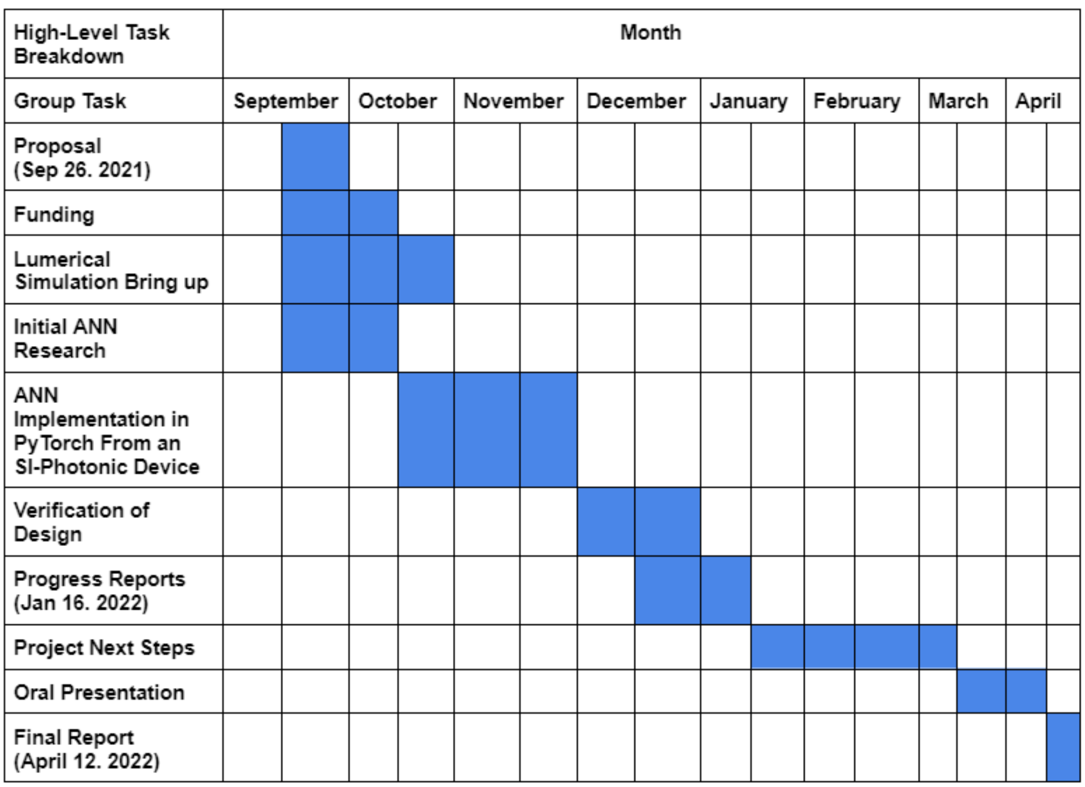

Figure 7: Overall Project Timeline For the Academic Year

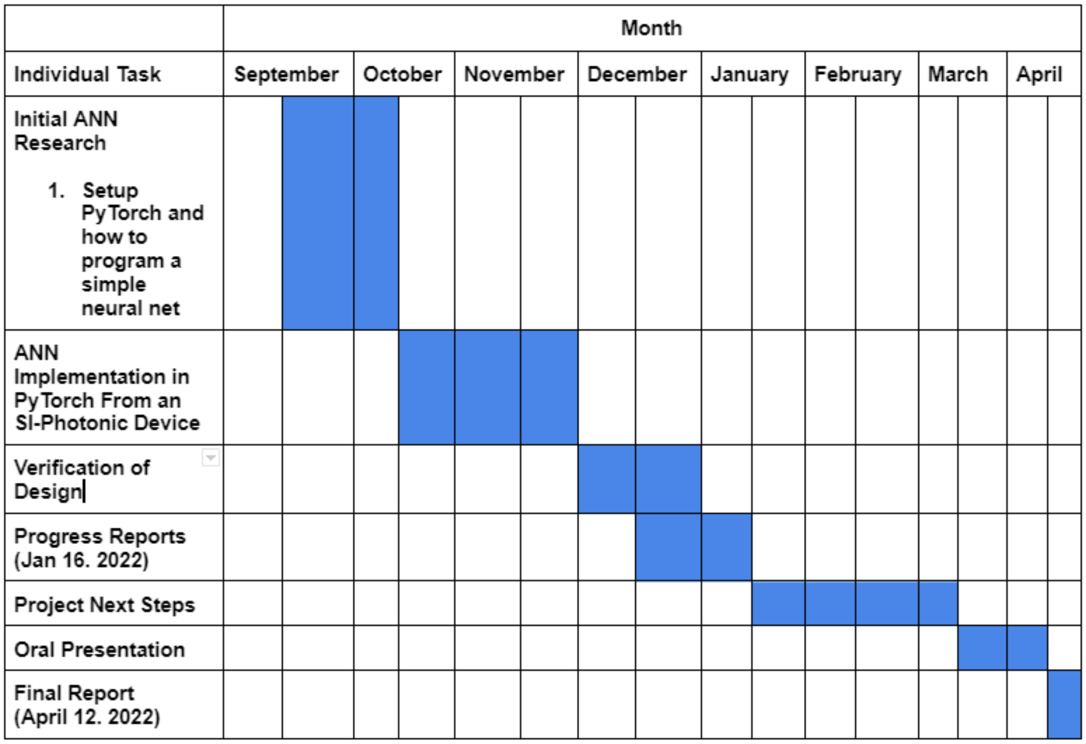

Figure 8: ANN Development Timeline For the Academic Year

3 Theory and Techniques

As outlined in the Introduction [1], the objective of this project is to develop an artificial neural network that can accurately model an SOI grating coupler so that we can find the optimum grating coupler parameters for a light ray at 1.55 μm.

The emission spectra of SOI grating couplers are highly dependent on their physical features and the features of the incident light rays. In the following section I explain the general approach to making an ANN to model a grating coupler and using that model to optimize the grating coupler parameters.

3.1 General Approach and Methods

The ANN structure I used is a multi-layer perceptron (as illustrated in Figure [4]). The weights in the ANN are initialized with a mean of zero and a standard deviation of one. Because the modelling of a grating coupler is a regression problem, we set the activation function at each node as the ReLU activation function; however, we omit the activation function at the output layer.

I evaluated the performance of the model by calculating the mean squared error loss function. The mean squared error is given by:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2 \tag{1}$$

where n is the number of data points, Y_i are the expected values and $\hat{Y}_i$ are the predicted values from the ANN model.

The loss function is minimized by mini-batch stochastic-gradient-descent (SGD). Mini-batch SGD splits the training set into smaller subsets and iteratively computes the loss function after each subset is passed into the ANN model. After each subset is passed into the ANN model the loss function is calculated and the weights are adjusted throughout the network accordingly with the Adam optimizer.

The Python code snippet on the next page was the routine I used to train the model *GratingCouplerNet* using mini-batch stochastic-gradient-descent. The **DataLoader** function is from the package **torch.utils.data** and it is used to split the training set into batches of size set by **batch_size**.

```python
Dataset = GratingCouplerDataset(x, y)
dataloader = DataLoader(dataset = Dataset, batch_size=10000)
learning_rate = 0.001
optimizer = torch.optim.Adam(GratingCouplerNet.parameters(),
            lr=learning_rate)
for epoch in range(max_epoch):
    for i, (x_train, y_train) in enumerate(dataloader):
        test_prediction = GratingCouplerNet(x_test)
        # Evaluate the training MSE loss from the Training set
        prediction = GratingCouplerNet(x_train)
        training_mse_error = mse_loss(prediction, y_train)
        # Evaluate the testing loss from the Testing set
        testing_mse_error = mse_loss(test_prediction, y_test)
        # Zero the gradients in the Network
        optimizer.zero_grad()
        # Update the weights and step the optimizer
        training_mse_error.backward()
        optimizer.step()
```

The object **GratingCouplerDataset** creates a dataset object and the class definitions are shown in the next page.

```python
class GratingCouplerDataset(torch.utils.data.Dataset):
    def __init__(self, x, y):
        self.x = torch.tensor(x, dtype=torch.float32)
        self.y = torch.tensor(y, dtype=torch.float32)
        self.length = self.x.shape[0]
    def __getitem__(self, idx):
        return self.x[idx], self.y[idx]
    def __len__(self):
        return self.length
```

I found that using mini-batch SGD offered better convergence performance than stochastic-gradient-descent or batch gradient descend.

The rate of gradient descent is controlled by the learning rate which is a hyperparameter for the Adam optimizer.

The specific ranges for the grating coupler parameters were: fiber angle ($5° - 20°$), SWG fill factor (0.2-0.6), duty cycle (0.4-0.8) and the pitch (0.5μm - 1.5μm). The modes of the incident light beams were the transverse electric TE and the transverse magnetic TM and the wavelength of the light ray was sampled from 1.3μm to 1.7μm.

There are six inputs (features) into the ANN and they consist of the fiber angle, pitch, duty cycle, pitch, light polarization and light wavelength and the output of the ANN model is the transmission value.

I then used the trained ANN to find the the grating coupler features (fiber angle, pitch, duty cycle) that resulted in the maximum amount of transmission at 1.55 μm.

3.2 Hardware and Software Tools

Today there are many software tools for implementing neural networks, even MATLAB has developed there own deep-learning toolbox!

We decided on using Python's PyTorch ML library for a few reasons: firstly, PyTorch has become one of the standard libraries for deep learning and neural network development, secondly we are very familiar with Python and lastly PyTorch is open source and so there is a plethora of support for it and it would enable us to share the project and continue with it more easily.

I used my home computer (shown in Figure [9] below) for the ANN training and development which was done on the CPU which is a 4.0 GHz 4-Core Intel Core i7 6700k and the RAM is 2400 MHz 64 Gb of DDR4 memory.

Figure 9: My Home Super Computer [Jonathan Levine]

4 Results and Discussion

For all the python files and the datasets see my project repository at: `https://github.com/JonathanALevine/Capstone`.

From datasets provided by the data acquisition team I combined two datasets of simulated data from Lumerical to create the training set.

One dataset consisted of parameterizing each parameter into 6 steps and the other into 8 steps. For example since the range of the fiber angle is $5° - 20°$, if split into 6 steps we get values of $5°$, $8°$, $11°$, $14°$, $17°$, $20°$. I also added more examples of simulation data to the training set beyond the 6-step and 8-step sets.

The training set that consisted of 11202 examples of fiber angle, pitch, duty cycle, fill factor, and for each example there was a 200 point transmission spectrum for both the TE and TM modes. The parameter ranges were: fiber angle ($5° - 20°$), SWG fill factor (0.2-0.6), duty cycle (0.4-0.8) and the pitch (0.5μm - 1.5μm). The modes of the incident light beams were the transverse electric TE and the transverse magnetic TM and the wavelength of the light ray was sampled from 1.3μm to 1.7μm. This resulted in a training set with 2,240,400 rows.

Similarly, the testing set consisted of 512 examples of fiber angle, pitch, duty cycle, fill factor, and for each example there was a 200 point transmission spectrum for both the TE and TM modes. The parameter ranges were: fiber angle ($10° - 20°$), SWG fill factor (0.2-0.6), duty cycle (0.4-0.8) and the pitch (0.5μm - 1.5μm). The modes of the incident light beams were the transverse electric TE and the transverse magnetic TM and the wavelength of the light ray was sampled from 1.3μm to 1.7μm. This resulted in a testing set with 102,400 rows.

We chose to narrow down the range of the fiber angle from $5° - 20°$ to $10° - 20°$ because we noticed that there are few examples where the grating coupler is able to couple light for fiber angles between $5° - 10°$. An example of a plot where the grating coupler fails to couple light is in Figure [10] below. In Figure [10] we can see that the grating coupler fails to couple light because the transmission values are ≈ 0.

The features of the training set (fiber angle, pitch, duty cycle fill factor, mode, wavelength) were all normalized using a min-max normalization and the label (transmission value) was normalized by taking the log of the value. The testing set was normalized on the same scale as the training set.

The training and testing sets were then used to train three ANN models of differing sizes and the results are discussed in the next section.

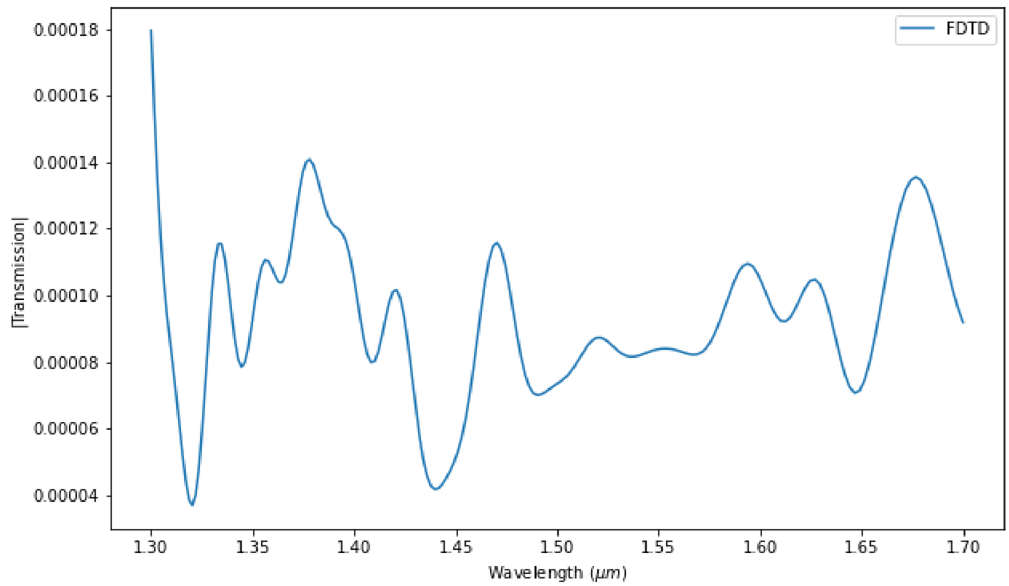

Figure 10: TM Mode Transmission Spectra of SWG Grating Coupler. Grating Coupler Parameters: Fiber Angle = 8.5°, Pitch = 0.886 μm, Duty Cycle = 0.8, Fill Factor = 0.6 [Jonathan Levine]

4.1 Results

The training times of ANN models of four, five and six hidden layers are shown Figure [11. As expected, in Figure [11] we can see that as the size of the ANN model increases, the training time also increases significantly. The training time for a six layer network is nearly double the training time for a four layer network.

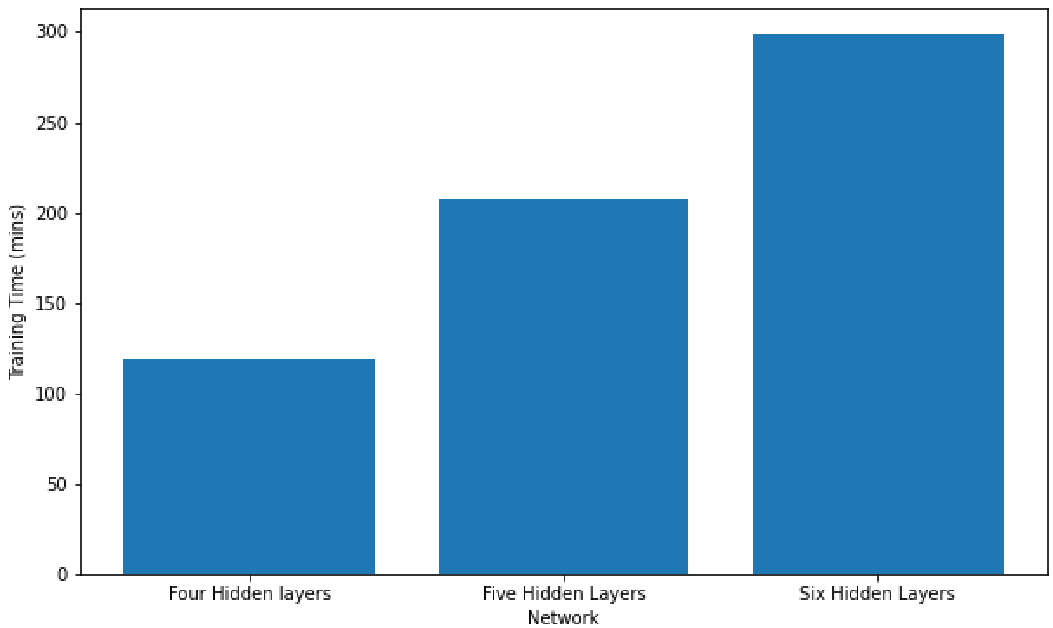

Figure 11: Training Time For Four, Five, and Six Hidden Layer Networks [Jonathan Levine]

As shown in Figure [12] below, the learning rate was held at 0.001 for the entirety of the training cycle for all three models. A learning rate of 0.001 resulted in the best performance in terms of convergence of the models. I found that higher learning rates prevented the models from converging to a solution. I also found that smaller learning rates resulted very noisy loss values.

In Figure [13] below we can see that the six-hidden-layer network performed the best (has the smallest MSE). Figure [14] illustrates the structure of the ANN six-hidden-layer model.

In figure [14] the inputs are illustrated by the blue circles, the hidden layers are illustrated by the orange boxes where the number of nodes is written inside, and finally, the output layer is illustrated by the green circle.

In the next section I show a validation example of the spectrum generated by the ANN model compared to what was generated from Lumerical.

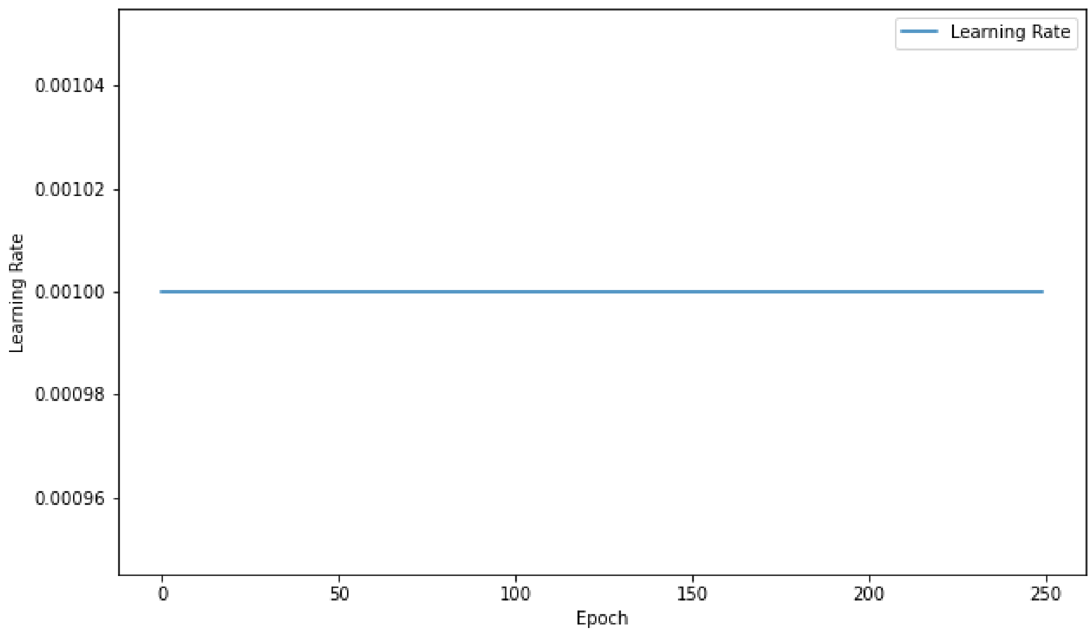

Figure 12: Learning Rate Over Epochs [Jonathan Levine]

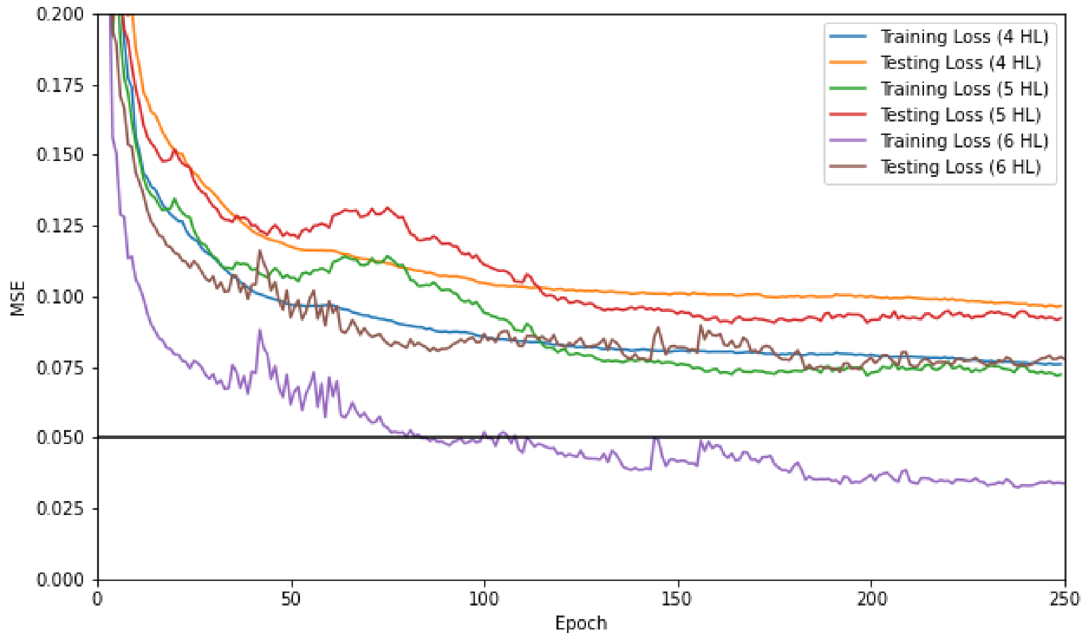

Figure 13: Training Losses [Jonathan Levine]

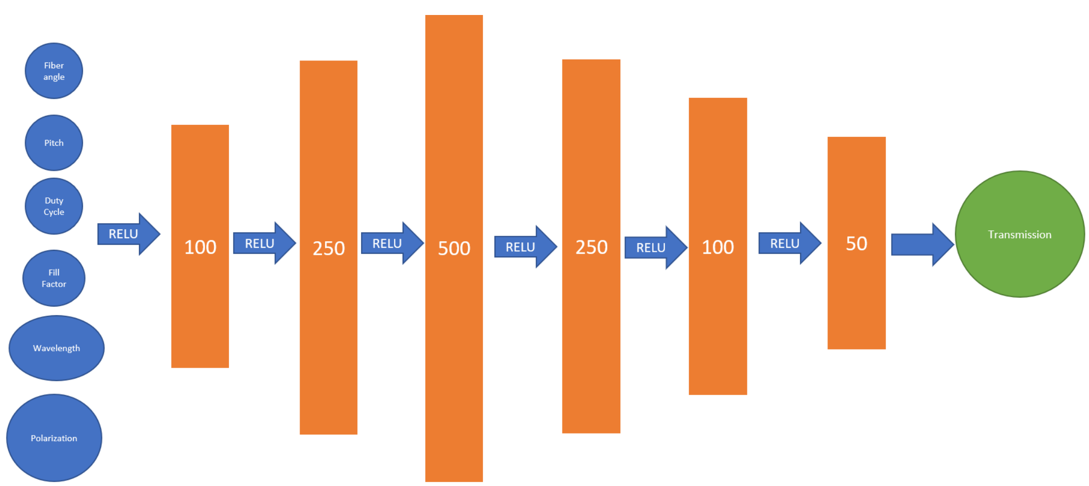

Figure 14: ANN Diagram [Jonathan Levine]

4.1.1 Validation Example

Figures [15] and [16] below show a validation example where the grating coupler parameters were set to: Fiber Angle = 14.375°, Pitch = 0.75 μm, Duty Cycle = 0.65 and Fill Factor = 0.25.

The inference time of the ANN for the 200-point transmission spectrum was about 0.121 seconds while Lumerical took about 130 seconds to complete the simulation. This results in about a 1074 times improvement in simulation time between the ANN model and FDTD simulations.

In Figures [15] and [16] we can see that the ANN model produces some noise at the maximums of the curves. I was not able to get my ANN models to filter out those edges, but myself and the rest of the group believe they might be related to specific causes. I discuss them in the Conclusion (Section [5.2]).

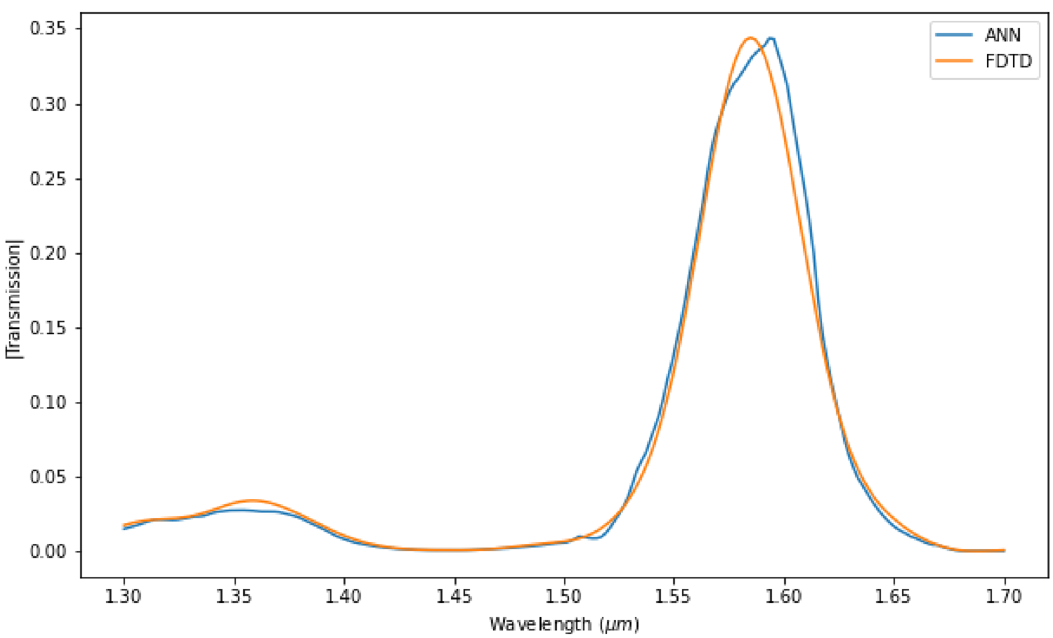

Figure 15: TE Mode Spectrum For Validation Example [Jonathan Levine]

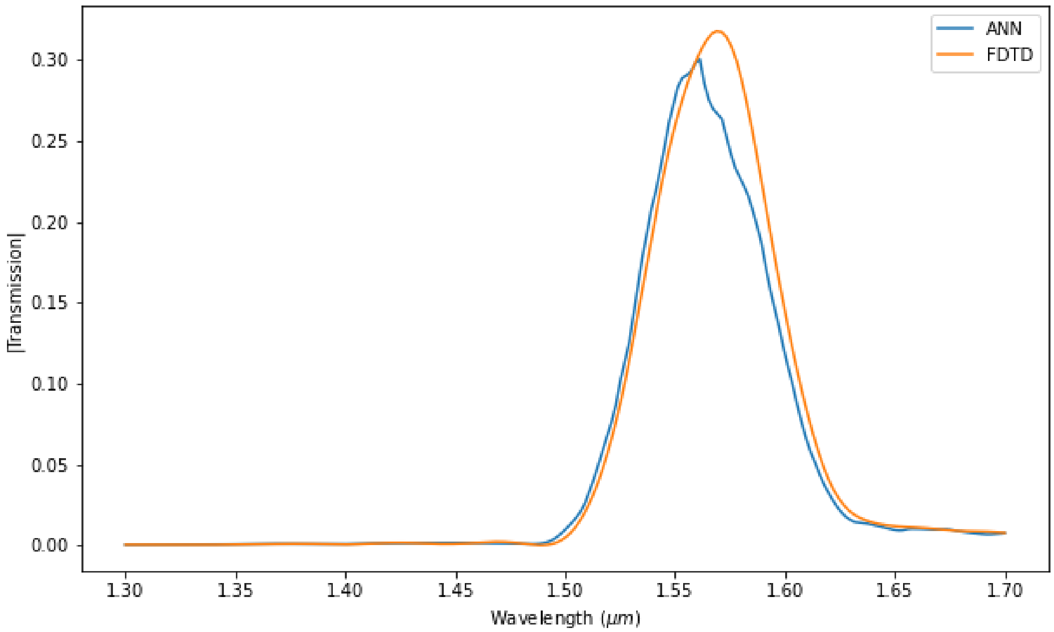

Figure 16: TM Mode Spectrum For Validation Example [Jonathan Levine]

4.1.2 Optimizing Parameters of a Grating Coupler

The trained ANN model, (GratingCouplerNet), was then used to find the optimum grating coupler parameters that give the maximum peak transmission value at a wavelength of 1.55 μm in the TE mode.

I found the optimum parameters by parameterizing each of the grating coupler features (fiber angle, pitch, duty cycle, fill factor) into ten steps and then calculating the transmission value. This resulted in $10^4 = 1000$ simulations that the ANN model had to run. Figure [17] below is a code snippet from my Jupyter Notebook where I use my ANN model (GratingCouplerNet) to find the optimum grating coupler parameters.

```python
# Set up the parameters for a sweep
# Optimize the transmission for the TE mode at 1.55 um
start_time = time.time()
thetas = np.linspace(dataframe.min()['Theta'], dataframe.max()['Theta'], 10)
pitchs = np.linspace(dataframe.min()['Pitch'], dataframe.max()['Pitch'], 10)
duty_cycles = np.linspace(dataframe.min()['Duty Cycle'], dataframe.max()['Duty Cycle'], 10)
fill_factors = np.linspace(dataframe.min()['Fill Factor'], dataframe.max()['Fill Factor'], 10)
mode = 0
wavelength = 1.55*10**(-6)

max_val = 0;
params = [0, 0, 0, 0]
mode = 0
count = 1

for theta in thetas:
    for pitch in pitchs:
        for duty_cycle in duty_cycles:
            for fill_factor in fill_factors:
                x = get_normalized_data_point(theta, pitch, duty_cycle, fill_factor, wavelength, mode)
                if max_val < get_transmission_val(GratingCouplerNet(x)):
                    max_val = get_transmission_val(GratingCouplerNet(x))
                    params[0] = theta
                    params[1] = pitch
                    params[2] = duty_cycle
                    params[3] = fill_factor
                count = count + 1
end_time = time.time()
print(params, max_val)
print(end_time-start_time)
```

```
[13.333333333333334, 7.244444444444444e-07, 0.5777777777777778, 0.5111111111111111] [0.40614742]
2.9683890342712402
```

Figure 17: Sweep For Finding Optimized Grating Coupler Parameters [Jonathan Levine]

The ANN model took approximately 2.96 minutes to complete the above calculations and found that grating coupler parameters of fiber angle = 13.33°, Pitch = 0.724 μm, Duty Cycle = 0.578 and Fill Factor = 0.511 gave the best transmission value at 1.55 μm. The ANN then generated the transmission spectrum for both the TE and the TM modes as shown in Figures [18] and [19] below.

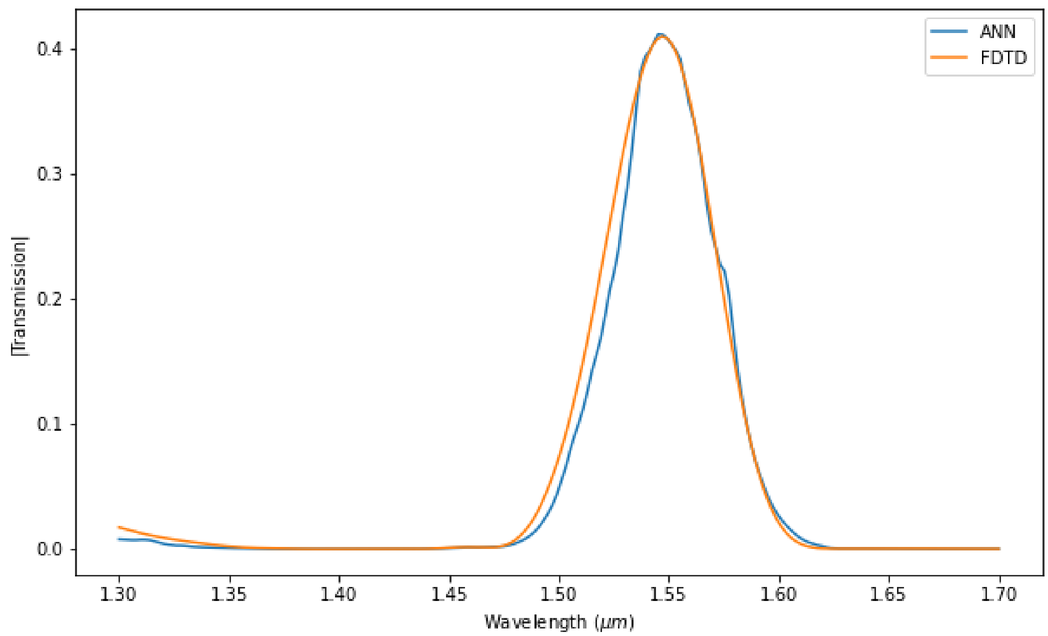

Figure 18: TE Mode Transmission Spectrum For Optimized grating Coupler Parameters [Jonathan Levine]

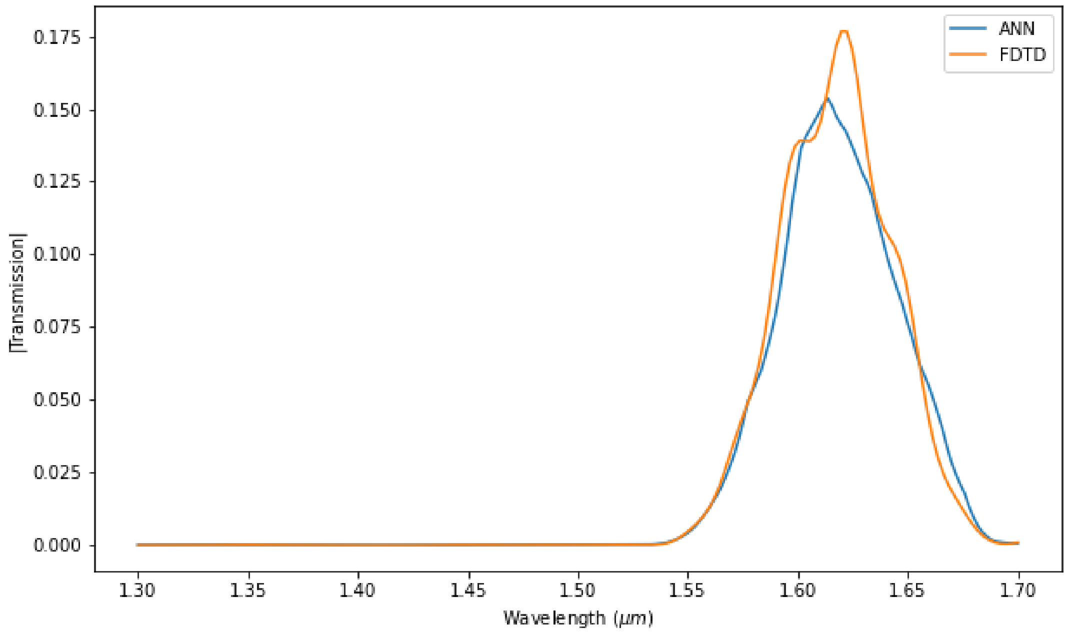

Figure 19: TM Mode Transmission Spectrum for Optimized Grating Coupler Parameters [Jonathan Levine]

4.1.3 Limits Of The ANN Model

As I previously mentioned at the beginning of this section, we noticed that grating couplers have trouble coupling light for fiber angles between $5° - 10°$. This means that in my training set there are few examples where the grating coupler produces a smooth curve in this fiber angle range.

The result of this is that the ANN model does not perform well at this fiber angle range as shown in Figures [20] and [21] below. The grating coupler parameters were set to: Fiber Angle = $5°$, Pitch = 0.7 μm, Duty Cycle = 0.48 and Fill Factor = 0.2.

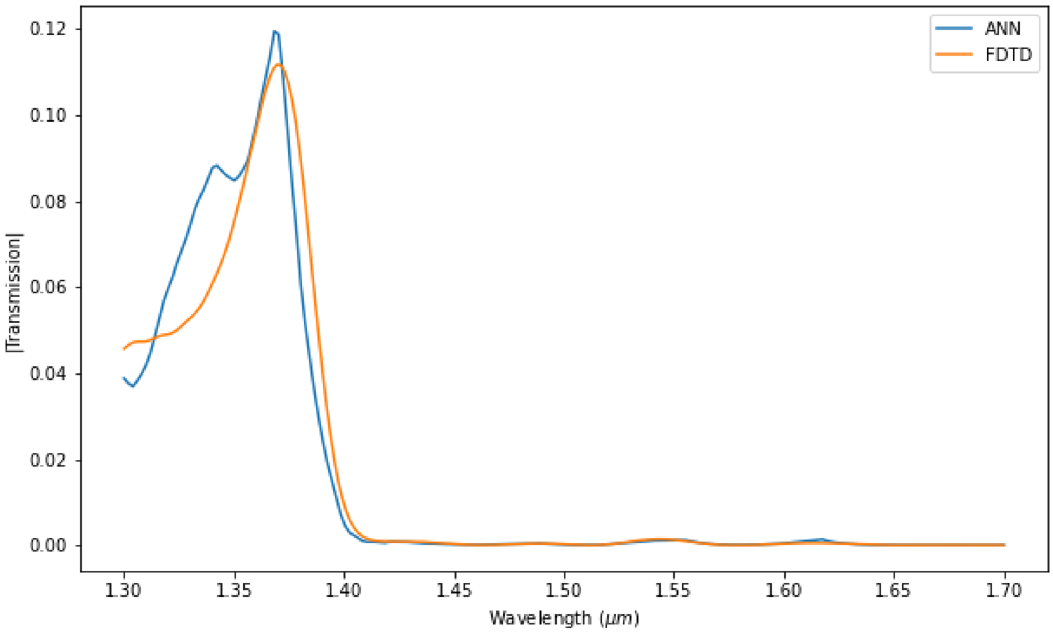

Figure 20: TE Mode [Jonathan Levine]

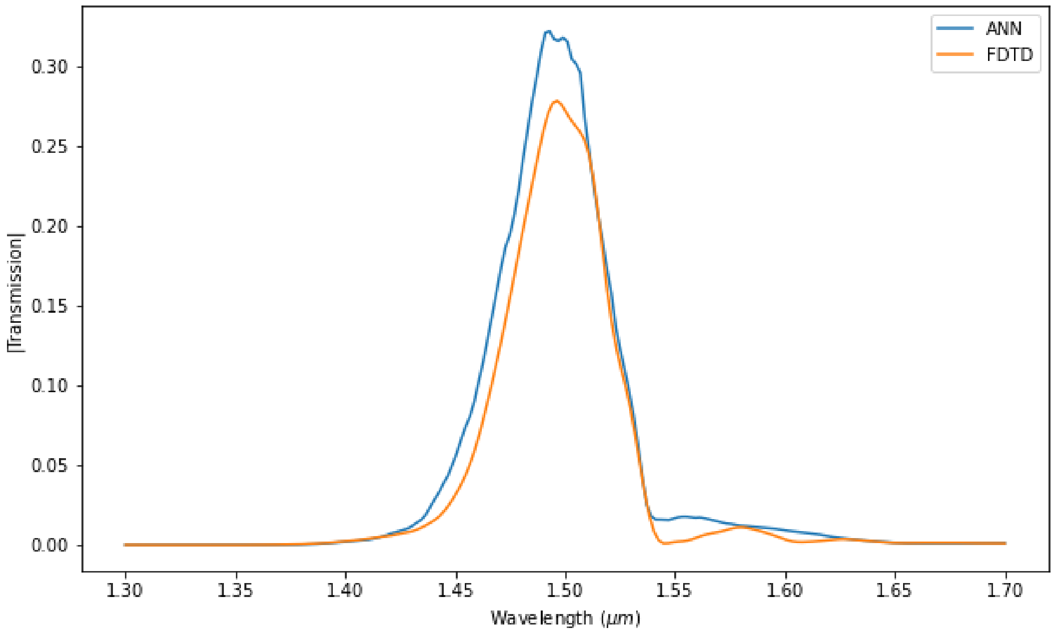

Figure 21: TM Mode [Jonathan Levine]

5 Conclusion

5.1 Summary and Contributions

In conclusion, throughout the past academic year I designed a working ANN model that simulates the transmission spectrum of a silicon-on-insulator grating coupler.

As a group we automated the process of generating simulation data from Lumerical and formatting that data into usable datasets for our ANN model development.

My contributions to the project were that I developed an ANN model that was able to match the transmission spectrum generated from FDTD simulations in Lumerical. The ANN model drastically sped up the simulation time of a grating coupler. The inference time of the ANN model on a 200-point transmission spectrum was 1074 times faster than the FDTD simulation from Lumerical.

I also showed a use case for an optical designer where they could use the ANN model to find the optimum parameters of a grating coupler for the TE mode at a wavelength of light of 1.55 μm.

5.2 Final Remarks and Future Work

From the results section we saw that the ANN model produced some noise at the maximum of the transmission spectrum curves. I suspect this is because in the training sets, there are examples of transmission spectra where the grating coupler fails to couple light; we saw this in Figure [10]. This would cause some noise in the output of the ANN because the weights throughout the network are being set to fit both a smooth Gaussian-like curve like in Figure [3] and highly-polynomial curves like in Figure [10]. The next step would be to filter out combinations of fiber angle, pitch, duty cycle and fill factor from the training and testing sets that result in highly-polynomial transmission spectra.

Secondly, I found the optimum learning rate, number of hidden layers and number of nodes per layer for the ANN model by trial and error. I recently discovered PyTorch's implementation of hyper-parameter tuning [6]. In this we can specify a search space for all our hyper-parameters (learning rate, hidden layers, node count, etc.) and programmatically find the combinations that result in our models having the best performance.

Thirdly, it is not yet clear how big a training set and testing set is necessary to fully capture the behaviour of an SOI grating coupler. I think it would be interesting to look into an iterative approach to generating training and testing sets. We could start with a relatively small training and testing set and then using those sets to train an ANN model. Then we would pick a random set of fiber angle, pitch, duty cycle and fill factor and compare the transmission spectra from the ANN and from Lumerical. If the MSE error is large we could add that example to the training set. We would then do this process iteratively till we reach an accuracy that we like. I illustrated this approach in Figure [22] below.

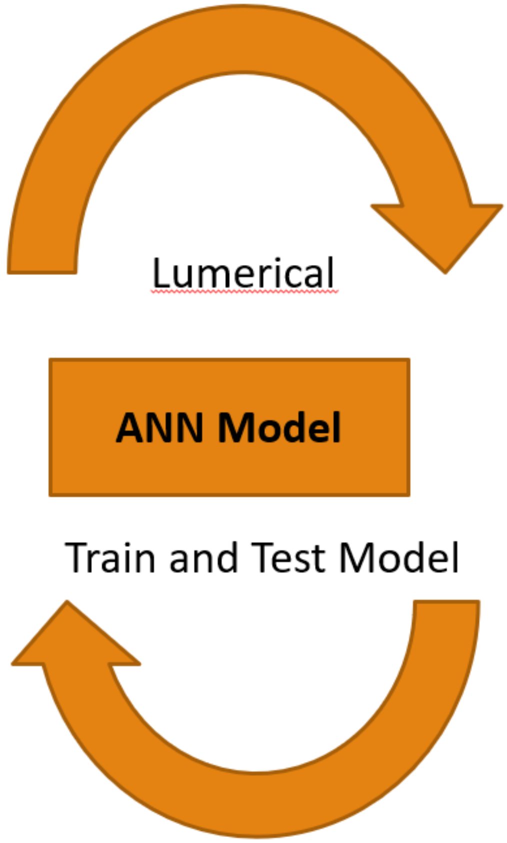

Figure 22: Iterative Approach To Generating Datasets [Jonathan Levine]

In conclusion I think the project was a success and there are many avenues to continue on with the project. For example, there is a growing focus in the STEM field of developing greener and more energy efficient technologies and I think it would be interesting to compare the energy usage between running simulations in Lumerical and running simulations on an ANN model.

References

[1] D.Gostimirovic and W.N.Ye, ”An Open-Source Artificial Neural Network Model for Polarization-Insensitive Silicon-on-Insulator Subwavelength Grating Couplers,” ***IEEE** JOURNAL OF SELECTED TOPICS IN QUANTUM ELECTRONICS, VOL, 25, NO.3, MAY/JUNE 2019.*

[2] (2022, Apr. 5). “Polarization-independent grating coupler based on silicon-on-insulator” [Online]. Available: `https://www.researching.cn/articles/OJc2ecdbdc7da0bab2/figureandtable`

[3] (2022, Apr. 5). “Lumerical” [Online]. Available: `https://www.lumerical.com/`

[4] (2021, Sept. 25). “Efficient Modeling of Complex Analog Integrated Circuits Using Neural Networks” [Online]. Available: `https://www.researchgate.net/publication/305410960_Efficient_Modeling_of_Complex_Analog_Integrated_Circuits_Using_Neural_Networks`

[5] (2021, Sept. 25). “Designing Your Neural Networks” [Online]. Available: `https://www.google.com/url?sa=i&url=https%3A%2F%2Ftowardsdatascience.com%2Fdesigning-your-neural-networks-a5e4617027ed&psig=AOvVaw3eQb6L4YoFnAPSMdc1Ix6m&ust=1632703391203000&source=images&cd=vfe&ved=0CAsQjRxqFwoTCMiYnpu0m_MCFQAAAAAdAAAAABAN`

[6] (2022, April 5). “Hyperparameter Tuning of Neural Networks with Optuna and PyTorch” [Online]. Available: `https://towardsdatascience.com/hyperparameter-tuning-of-neural-networks-with-optuna-and-PyTorch-22e179efc837`