**BAB II**

**TINJAUAN PUSTAKA**

**2.1 Sifat Non-Linear Serat Optik**

Penelitian terhadap serat optik menjadi sangatlah fundamental guna meningkatkan kecepatan respons sinyal dalam sistem komunikasi optis. Faktor indeks pembiasan nonlinear serat optik memiliki kaitan yang erat dalam penelitian tersebut. Hal ini dilatarbelakangi oleh perkembangan jaringan optis pada tahun 1970, di mana besar perlemahan sinyal berhasil ditekan di bawah 20 dB/km. Komunikasi berbasis serat optik kemudian dapat dipergunakan secara luas dengan biaya yang lebih terjangkau. (Kah et al., 2022).

Sifat nonlinearitas dari serat optik digunakan dalam menjelaskan beberapa fenomena optis. Sebagai contoh, hilangnya sebagian sinyal oleh karena distribusi tak seragam dari intensitas pulsa, dan pembentukan soliton melalui modulasi fase dan kecepatan dispersi grup (Abdillah Mardi et al., 2023). Soliton sendiri merupakan paket gelombang yang mempertahankan bentuknya saat merambat dalam medium nonlinier dispersif tanpa mengalami penyebaran. Pemodelan matematis dari perambatan cahaya, termasuk soliton, dalam serat optik ditunjukkan dalam persamaan diferensial parsial (PDE) 2.1

|  |  |
| --- | --- |
|  | (2.1) |

(Jiang et al., 2022)

menyatakan fungsi gelombang sebagai solusi 2.1 sementara dan masing-masing menyatakan daya atenuasi, koefisien dispersi, dan nonlinearitas indeks bias. Ketiga parameter ini bertanggung jawab atas karakteristik perambatan sinyal dalam serat optik.

**2.2 Metode *Split Step Fourier***

Metode *Split-Step Fourier* (SSFM) menjadi metode yang paling umum digunakan dalam memberikan aproksimasi yang akurat dalam perambatan gelombang. Metode ini mengaproksimasikan efek nonlinear dan dispersi secara terpisah sehingga mampu mengadopsi efek nonlinear, statis, maupun efek stokastik dengan baik. Hal ini dicapai dengan memisahkan persamaan 2.1 menjadi operator linear dan nonlinear sebagaimana dalam persamaan 2.2 – 2.3.

|  |  |
| --- | --- |
|  | (2.2) |
|  | (2.3) |

(Farag et al., 2021)

Dengan menggunakan transformasi fourier pada kedua sisi, PDE dapat diubah menjadi persamaan diferensial biasa dalam domain frekuensi dalam persamaan 2.4, sehingga mendapatkan persamaan akhir 2.5.

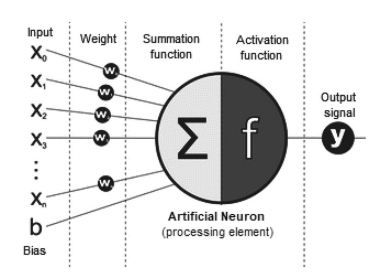
|  |  |
| --- | --- |
|  | (2.4) |
|  | (2.5) |

(Farag et al., 2021)

**2.4 Deep Learning**

*Artificial Neural Network* (ANN) merupakan model matematis dari saraf buatan yang dikembangkan pada 194.3. Dalam perkembangan ANN, konsep dari *Deep Learning* (DL) diperkenalkan pada tahun 2006 dan telah memberikan kontribusi yang signifikan bagi perkembangan teknologi dan ilmu pengetahuan. Pendekatan berbasis DL menawarkan solusi yang baik dalam membangun lingkungan sistem berbasis data. (Sarker, 2021; Talaei Khoei et al., 2023).

Sistem DL tersusun atas banyak unit pemrosesan tunggal, bernama neuron, yang saling terhubung. Masing-masing neuron akan menghasilkan serangkaian nilai variabel *weight* dan *bias*. Elemen yang terdapat dalam sebuah neuron ditunjukkan dalam Gambar 2.1 di mana nilai input () diporses berdasarkan nilai *weight* ( dan *bias* (*b)* melalui suatu fungsi linear 1. Hasil penjumlahan ini kemudian dilewatkan melalui suatu fungsi aktivasi non-linear, untuk mendapatkan nilai output (Sarker, 2021). Keseluruhan proses ini dinamakan dengan *forward propagation.,* dan dirumuskan dalam persamaan 2.1. Hasil dari nilai output akan dibandingkan dengan nilai sesungguhnya melalui suatu *cost function* yang menyatakan nilai loss, atau seberapa besar error model.

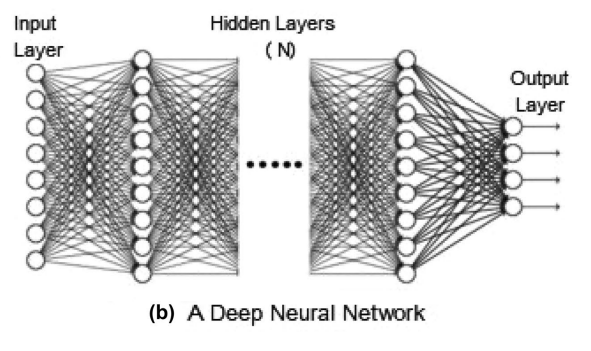


Gambar 2.1 Skema *Artificial Neuron*  (Sarker, 2021)

|  |  |
| --- | --- |
|  | (2.6) |

(Cuomo et al., 2022)

Pondasi dari arsitektur DNN merupakan sistem *Multi Layer Perceptron* (MLP). Sistem MLP tersusun atas sejumlah lapisan jaringan yang saling terhubung, berupa satu lapisan input, satu lapisan output, dan sebanyak lapisan tersembunyi di antara keduanya. Lapisan tersembunyi ini berfungsi sebagai mesin komputasi sistem, masing-masing tersusun atas sejumlah unit neuron dilengkapi dengan fungsi aktivasi.



Gambar 2.2 Multi Layer Perceptron (Sarker, 2021)

Semakin banyak lapisan neuron yang dideklarasikan, algoritma pembelajaran ANN menjadi semakin dalam. Tiap lapisan dapat dispesifikasikan untuk mempelajari aspek tertentu dalam suatu data. Murtagh, (1991) menyatakan bahwa banyaknya lapisan neuron, maupun banyaknya neuron dalam suatu lapisan dapat meningkatkan sensitivitas kalkulasi data, tetapi juga dapat meningkatkan resiko *overfitting*.

Proses optimalisasi nilai weight dan bias pada setiap neuron dilakukan melalui metode *backpropagation*. Metode ini memungkinkan setiap neuron belajar secara otomatis untuk mengenali fitur-fitur dari data masukan yang relevan dalam menghasilkan keluaran yang diinginkan. Algoritma pembelajaran bertujuan untuk meminimalkan nilai *cost function* dengan memperbarui bobot neuron berdasarkan hasil minimisasi tersebut (Shrestha & Mahmood, 2019).

Strategi pembelajaran dalam optimalisasi model MLP meliputi bagaimana pendekatan yang digunakan untuk melatih model berdasarkan sifat data. Terdapat dua jenis dari strategi pembelajaran meliputi *online learning* dan *batch learning*. *Online learning* memproses data secara berurutan dan memperbarui model setiap data, menciptakan model yang dinamis. Sementara itu, *batch learning* memerlukan keseluruhan data yang dikelompokkan dalam suatu *batch* sebleum pelatihan dimulai. Hal ini menghasilkan model yang lebih sederhana tetapi kurang fleksibel (Talaei Khoei et al., 2023).

Dengan demikian, dapat disimpulkan bahwa desain dari MLP ditentukan oleh hal-hal berikut: (a) arsitektur MLP, meliputi jumlah lapisan dan neuron pada masing-maisng lapisan; (b) fungsi aktivasi pada *hidden layer* MLP; (c) fungsi algoritma pembelajaran dalam *backpropagation*; (d) strategi pembelajaran yang digunakan dalam melatih model (Murtagh, 1991).

**2.4 Penurunan Gradien**

Penurunan gradien merupakan metode optimisasi yang digunakan guna meminimalisasi suatu fungsi , yang merupakan nilai loss model, terhadap sebagai parameter jaringan neuron. Optimasi dilakukan dengan memperbarui nilai pada arah yang berlawanan terhadap gradien . Parameter yang menyatakan seberapa besar langkah dari penurunan gradien sehingga mencapai titik minimum dinyatakan oleh nilai skalar , laju pembelajaran model.

Metode penurunan gradien dibagi menjadi tiga berdasarkan banyaknya data yang digunakan dengan mempertimbangkan akurasi dan waktu dalam perhitungan; *batch gradient descent, stochastic gradient descent, mini-batch gradient descent*. Pada *batch gradient descent*, pembaharuan parameter dilakukan untuk seluruh dataset setelah melakukan satu iterasi penuh. Hal ini membuat metode tersebut bisa menjadi sangat lambat dan membutuhkan memori yang besar dengan semakin banyaknya dataset. Metode ini, secara matematis dirumuskan dalam persamaan 2.7.

|  |  |
| --- | --- |
|  | (2.7) |

Sementara itu, dalam *stochastic gradient descent*, parameter diperbarui untuk tiap data dalam suatu dataset. Optimasi parameter, dengan demikian, menjadi lebih cepat dengan penggunaan memori yang lebih efisien. Akan tetapi, dengan semakin seringnya pembaharuan yang dilakukan dengan variansi *loss* yang cukup besar, parameter akan berubah dengan fluktuatif. Akan tetapi, fluktuasi SGD dapat diatasi dengan mengurangi laju pembelajaran , walau hal ini mengakibatkan konvergensi menjadi lebih lambat.

|  |  |
| --- | --- |
|  | (2.7) |

Alternatif dari kedua metode tersebut berupa *mini-batch gradient descent* di mana pembaharuan parameter dilakukan untuk setiap kelompok yang berisi data. Hal ini mengurangi variansi dari pembaruan parameter dengan kebutuhan memori yang tidak seberat *batch gradient descent.*

|  |  |
| --- | --- |
|  | (2.8) |

(Ruder, 2016).

**2.6 Physics-Instructed Neural Network (PINN)**

Physics-Instructed Neural Network merupakan suatu model ANN yang digunakan spesifik dalam menyelesaikan persamaan diferensial parsial. Arsitektur jaringan in berbeda dengan model DNN pada umumnya yang hanya memproses data input dan meminimalisasi nilai loss berdasarkan nilai sesungguhnya. PINN turut menaruh perhatian terhadap batasan fisis yang diberikan oleh persamaan diferensial serta syarat awal dan syarat batas. (Cuomo et al., 2022; Jiang et al., 2022). Diagram dari arsitektur PINN ditunjukkan pada Gambar 2.3

A diagram of a algorithm

Description automatically generated

Gambar 2.3 Arsitektur PINN (Jiang et al., 2022)

Gambar 2.3 menunjukkan bagaimana penyelesaian sebuah fungsi turunan = melalui PINN. Sistem PINN menerima input berupa *collocation point*; titik dalam suatu domain () yang dipilih sebagai representasi perhitungan dan titik data terlabel; titik ) di mana nilai telah diketahui (seperti syarat batas maupun syarat awal).

Susunan jaringan neuron digunakan untuk mendapatkan nilai fungsi melalui *forward propagation*. Dalam proses *forward propagation,* tiap neuron melakukan perhitungan linear berdasarkan nilai *weight* dan *bias* neuron, yang selanjutnya diberikan kepada suatu fungsi aktivasi. Cuomo et al., (2022) menyatakan bahwa pemilihan fungsi aktivasi memberikan dampak yang signifikan terhadap performa pembelajaran, dengan fungsi yang sering kali digunakab berupa sigmoid, tangensial hipernolik, ReLu, *leaky* ReLu, dan Swish.

Nilai fungsi yang didapatkan kemudian diturunkan terhadap variabel () sesuai dengan derajat turunan yang terdapat dalam persamaan turunan parsial .

Proses *backward propagation* guna mengoptimalisasi tiap *weight* neuron pertama-tama dilakukan dengan menginisialisasi nilai loss. Gambar 2.3 menunjukkan bahwa terdapat dua nilai loss berupa dan . menyatakan nilai loss dari data prediksi terhadap data terlabel , dinyatakan dalam persamaan 2.2. Sementara itu menyatakan nilai residu dari persamaan turunan .

|  |  |
| --- | --- |
|  | (2.3) |
|  | (2.4) |

(Cuomo et al., 2022)

Penjumlahan nilai loss selanjutnya diproses menggunakan fungsi optimalisasi dalam algoritma pembelajaran. Hasil pemrosesan nilai loss diumpanbalikkan ke tiap neuron untuk mengoptimalisasi nilai *weight* dan *bias*. Keseluruhan proses ini diulang terus menerus hingga mencapai *epoch*/iterasi tertentu sehingga didapatkan hasil yang konvergen.

**2.7 Deep Learning dalam Penyelesaian NLS**

**2.7.1 Data Driven Machine Learning**

Metode *Data Driven Machine Learning* dalam penyelesaian NLS memanfaatkan kemampuan umpan balik dari arsitektur *deep neural network.* Dalam memodelkan perambatan cahaya dalam serat linear optik, diperlukan data pembelajaran atas sinyal input dan sinyal output yang diketahui. Dengan demikian, sistem ANN akan mempelajari data yang diberikan untuk memprediksi sinyal output dari sinyal input yang lain. (Jiang et al., 2021).

Metode ini salah satunya ditunjukkan dalam Yang et al., (2021) di mana *generative adversarial network* (GAN) digunakan dalam memodelkan distribusi fungsi transfer serat optik. Penelitian ini memberikan metode yang jauh lebih efisien dari *Split Step Method* dengan kompleksitas yang lebih kecil. Akan tetapi, kelemahan dari metode GAN ditunjukkan atas bagaimana model hanya mempelajari sebagian fitur nonlinier data transmisi dan mengabaikan fitur lainnya. Hal ini menyebabkan gangguan pada keakuratan simulasi pada transmisi jarak jauh jika dibandingkan dengan *split step method*. Kelemahan ini turut ditunjukkan dalam Jiang et al., (2021) di mana *data driven machine learning* memerlukan sejumlah besar data dalam proses pembelajaran tanpa mengindahkan prinsip fisis dari data tersebut.

**2.7.2 Algoritma PINN Dalam Persamaan NLS**

Fungsi gelombang optis dari NLS dinyatakan dalam suatu fungsi real dan imajiner dalam fungsi 2.4.

|  |  |
| --- | --- |
|  | (2.5) |

Ditunjukkan dalam Gambar 2.4, model PINN akan memberikan output terhadap suatu fungsi dan , masing-masing merupakan fungsi real yang menyatakan komponen real dan imajiner, dan dari .

A diagram of a complex network

Description automatically generated

Gambar 2.4 Arsitektur PINN Dalam Penyelesaian NLS (Jiang et al., 2022)

Arsitektur PINN tersusun atas jaringan neural untuk menghitung nilai dan Sistem jaringan neural akan menerima *collocation point*, titik representatif dari dimensi dan beserta data terlabel dari syarat awal dan syarat batas. Turunan fungsi dihitung untuk menentukan error MSE, dengan dan sebagai residu fungsi real dan imajiner, serta dan sebagai nilai loss real dan imajiner terhadap data terlabel. Proses propagasi balik meminimalkan keempat unsur untuk mengoptimalkan bobot dan bias model Proses ini diulang hingga tercapai hasil yang konvergen dengan nilai *loss* minimum.

Optimisasi parameter dalam PINN, menurut Cuomo et al., (2022) dapat dilakukan menggunakan metode *mini-batch gradient descent* dengan algoritma L-BFGS-B (*limited-memory Broyden–Fletcher–Goldfarb–Shanno*), sebuah metode optimasi *quasi-Newton,* yang memungkinkan proses pembelajaran yang efisien dengan. Pendekatan ini menghasilkan konvergensi yang cepat dengan beban komputasi rendah pada ukuran sampel yang kecil. Namun, peningkatan ukuran sampel dan jumlah epoch pembelajaran dapat memberikan hasil yang lebih optimal. Sebaliknya, penggunaan *stochastic gradient descent* dilaporkan menghadapi kesulitan dalam menangani collocation points yang acak pada sistem 3D.

**2.8 TensorFlow**

A close up of a logo

Description automatically generated

Gambar 2.5 TensorFlow (TensorFlow Developers, 2024)

TensorFlow merupakan *framework* pembelajaran mesin dirancang untuk mendukung proses berskala besar pada lingkungan heterogen dengan beragam perangkat keras. Sistem TensorFlow mampu mendistribusikan komputasi pada berbagai perangkat komputasi seperti *multicore* CPU, GPU, dan TPU. Arsitektur ini memberikan fleksibilitas dalam pembelajaran dan inferensi dari algoritma jaringan neural. TensorFlow telah digunakan dalam berbagai layanan Google, dan telah menjadi telah umum digunakan dalam penelitian berbasis ML dan proyek *open-source* (Abadi et al., 2016).

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., … Zheng, X. (2016). TensorFlow: A system for large-scale machine learning. *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2016*, 265–283.

Abdillah Mardi, H., Nasaruddin, N., Ikhwan, M., Nurmaulidar, N., & Ramli, M. (2023). Soliton dynamics in optical fiber based on nonlinear Schrödinger equation. *Heliyon*, *9*(3), e14235. https://doi.org/10.1016/j.heliyon.2023.e14235

Cuomo, S., Di Cola, V. S., Giampaolo, F., Rozza, G., Raissi, M., & Piccialli, F. (2022). Scientific Machine Learning Through Physics–Informed Neural Networks: Where we are and What’s Next. *Journal of Scientific Computing*, *92*(3), 1–62. https://doi.org/10.1007/s10915-022-01939-z

Farag, N. G. A., Eltanboly, A. H., EL-Azab, M. S., & Obayya, S. S. A. (2021). On the Analytical and Numerical Solutions of the One-Dimensional Nonlinear Schrodinger Equation. *Mathematical Problems in Engineering*, *2021*(1). https://doi.org/10.1155/2021/3094011

Jiang, X., Wang, D., Chen, X., & Zhang, M. (2022). Physics-Informed Neural Network for Optical Fiber Parameter Estimation From the Nonlinear Schrödinger Equation. *Journal of Lightwave Technology*, *40*(21), 7095–7105. https://doi.org/10.1109/JLT.2022.3199782

Jiang, X., Wang, D., Fan, Q., Zhang, M., Lu, C., & Lau, A. P. T. (2021). Solving the nonlinear Schrödinger equation in optical fibers using physics-informed neural network. *Optics InfoBase Conference Papers*, 1–3. https://doi.org/10.1364/ofc.2021.m3h.8

Kah, P. M., Kiogora, P. R., Awuor, K., & Saoke, C. (2022). Fast Solution for the Nonlinear Schrodinger Equation in Optical Fibers by the Reduced Basis Method. *International Journal of Mathematics Trends and Technology*, *68*(1), 101–114. https://doi.org/10.14445/22315373/ijmtt-v68i1p512

Murtagh, F. (1991). Multilayer perceptrons for classification and regression. *Neurocomputing*, *2*(5–6), 183–197. https://doi.org/10.1016/0925-2312(91)90023-5

Ruder, S. (2016). *An overview of gradient descent optimization algorithms*. 1–14. https://doi.org/10.48550/arXiv.1609.04747

Sarker, I. H. (2021). Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. *SN Computer Science*, *2*(6), 1–20. https://doi.org/10.1007/s42979-021-00815-1

Talaei Khoei, T., Ould Slimane, H., & Kaabouch, N. (2023). Deep learning: systematic review, models, challenges, and research directions. *Neural Computing and Applications*, *35*(31), 23103–23124. https://doi.org/10.1007/s00521-023-08k957-4

TensorFlow\_Developers. (2024). *TensorFlow*. Zenodo. https://doi.org/10.5281/zenodo.13989084

Yang, H., Niu, Z., Xiao, S., Fang, J., Liu, Z., Fainsin, D., & Yi, L. (2021). Fast and Accurate Optical Fiber Channel Modeling Using Generative Adversarial Network. *Journal of Lightwave Technology*, *39*(5), 1322–1333. https://doi.org/10.1109/JLT.2020.3037905