**Jurnal Penelitian Teknologi Informasi dan Sains**
**Vol. 2 No. 3 September 2024**
e-ISSN :2985-7635, p-ISSN :2985-6280, Hal 59-69
DOI: https://doi.org/10.54066/jptis.v2i3.2358
*Available online at:* https://jurnal.itbsemarang.ac.id/index.php/JPTIS/

# Mengejar Kinerja Maksimal: Teknik Pengoptimalan Terkini dalam Pembelajaran Mesin

**Agung Yuliyanto Nugroho[1] Ferat Kristanto[2]**
*[1] Universitas Cendekia Mitra Indonesia
[2]SMK Telkom Purwokerto, Indonesia
*email: agungyuliyanto@unicimi.ac.id, feratk@smktelkom-pwt.sch.id*

*Korespondensi Penulis : agungyuliyanto@unicimi.ac.id*

***Abstract*** *In the era of data revolution and artificial intelligence, machine learning model optimization has become one of the most dynamic and crucial research areas. This article reviews the latest techniques in machine learning model optimization with a focus on pursuing maximum performance. We discuss various methods applied to improve model accuracy and efficiency, ranging from hyperparameter tuning techniques, advanced optimization algorithms such as Bayesian optimization, to innovative approaches such as meta-learning and transfer learning. These optimization techniques not only aim to improve model performance but also to overcome challenges related to big data, model complexity, and computational limitations. We investigate how these methods can be integrated in machine learning pipelines to achieve better results with more efficient resources. Through a review of recent literature and case studies of applications in various domains, this article provides in-depth insights into the trends and developments in model optimization, as well as practical recommendations for researchers and practitioners in pursuing maximum performance from their machine learning systems. A better understanding of these cutting-edge techniques is expected to facilitate the achievement of better and more innovative results in future machine learning applications.*

***Keyword*** *: Tuning Hiperparameter;Bayesian Optimization;Meta-Learning;Transfer Learning*

**Abstrak:** Dalam era revolusi data dan kecerdasan buatan, pengoptimalan model pembelajaran mesin (machine learning) telah menjadi salah satu area penelitian yang paling dinamis dan krusial. Artikel ini mengkaji teknik-teknik terbaru dalam pengoptimalan model pembelajaran mesin dengan fokus pada upaya mengejar kinerja maksimal. Kami membahas berbagai metode yang diterapkan untuk meningkatkan akurasi dan efisiensi model, mulai dari teknik tuning hiperparameter, algoritma optimasi canggih seperti Bayesian optimization, hingga pendekatan inovatif seperti meta-learning dan transfer learning. Teknik optimasi ini tidak hanya bertujuan untuk meningkatkan performa model, tetapi juga untuk mengatasi tantangan yang terkait dengan data besar, kompleksitas model, dan keterbatasan komputasi. Kami menyelidiki bagaimana metode-metode ini dapat diintegrasikan dalam pipeline pembelajaran mesin untuk mencapai hasil yang lebih baik dengan sumber daya yang lebih efisien. Melalui tinjauan literatur terbaru dan studi kasus aplikasi di berbagai domain, artikel ini memberikan wawasan mendalam tentang tren dan perkembangan dalam pengoptimalan model, serta rekomendasi praktis untuk para peneliti dan praktisi dalam mengejar kinerja maksimal dari sistem pembelajaran mesin mereka. Dengan pemahaman yang lebih baik tentang teknik-teknik terkini ini, diharapkan dapat memfasilitasi pencapaian hasil yang lebih baik dan inovatif dalam aplikasi pembelajaran mesin di masa depan.

**Kata Kunci:** *Tuning Hiperparameter; Bayesian Optimization; Meta-Learning; Transfer Learning*

## 1. PENDAHULUAN

### Definisi Pengoptimalan Model

Pengoptimalan model pembelajaran mesin adalah bagian penting dalam memahami bagaimana cara meningkatkan performa model dalam machine learning (pembelajaran mesin). Pengoptimalan model melibatkan teknik dan strategi untuk memilih dan menyetel parameter model agar mencapai hasil terbaik. (Friedman, J. H. (2001). Greedy function approximation:

*Received: Juni 27, 2024; Revised: Juli 25, 2024; Accepted: Agustus 21, 2024; Online Available: Agustus 23, 2024*

A gradient boosting machine. The Annals of Statistics, 29(5), 1189-1232. https://doi.org/10.1214/aos/1013203451 ).

Optimasi dalam pembelajaran mesin adalah proses mencari parameter model yang terbaik untuk meminimalkan atau memaksimalkan fungsi objektif, biasanya fungsi kerugian atau cost function. Tujuannya adalah untuk menemukan konfigurasi parameter yang menghasilkan performa terbaik pada data yang digunakan.

Fungsi objektif adalah metrik yang ingin Anda optimalkan. Dalam konteks pembelajaran mesin, ini sering berupa fungsi kerugian yang mengukur seberapa baik model memprediksi hasil yang benar. Contoh fungsi kerugian termasuk Mean Squared Error (MSE) untuk regresi atau Cross-Entropy Loss untuk klasifikasi.

## 2. METODE OPTIMASI UMUM

Metode optimasi umum dalam pembelajaran mesin adalah teknik yang digunakan untuk mengoptimalkan model dengan tujuan meminimalkan atau memaksimalkan fungsi objektif, biasanya berupa fungsi kerugian. Berikut adalah beberapa metode optimasi umum yang sering digunakan dalam pelatihan model pembelajaran mesin.

**Gradient Descent (GD)**

Gradient Descent adalah metode dasar dalam optimasi yang digunakan untuk memperbarui parameter model dengan menghitung gradien dari fungsi kerugian. Prosesnya melibatkan:

a. Batch Gradient Descent:

   Menggunakan seluruh dataset untuk menghitung gradien dan memperbarui parameter model. Ini bisa sangat lambat untuk dataset besar karena memerlukan pemrosesan data lengkap dalam setiap iterasi.

b. Stochastic Gradient Descent (SGD)

   Menggunakan satu contoh data untuk menghitung gradien dan memperbarui parameter. Ini mempercepat pembaruan tetapi dapat menghasilkan fluktuasi besar dalam fungsi kerugian.

c. Mini-Batch Gradient Descent

   Menggunakan subset data (mini-batch) untuk menghitung gradien dan memperbarui parameter. Ini adalah kompromi antara batch dan SGD, sering kali lebih efisien dan stabil.

**Momentum**

Momentum adalah metode yang mempercepat konvergensi dengan mempertimbangkan gradien dari iterasi sebelumnya. Ini membantu model untuk melanjutkan dalam arah yang sama dan mengurangi fluktuasi: (Sada, 2022).

**Tabel 1 : Rumus Momentum**

| No | Formula | Update |
|---|---|---|
| 1 | v t+1 =βv<br>t +(1−β)∇L(θ t ) | $\theta t+1=\theta t-\eta vt+1$<br>θ t+1 =θ t −ηv t+1 |

**1.** Nesterov Accelerated Gradient (NAG)

NAG adalah perbaikan dari Momentum yang lebih baik dalam menangani overshooting. Ini melakukan perkiraan langkah depan sebelum menghitung gradien:

- **Formula**: θ~=θt−βvt\tilde{\theta} = \theta_t - \beta v_tθ~=θt−βvt
- **Update**: vt+1=βvt+∇L(θ~)v_{t+1} = \beta v_t + \nabla L(\tilde{\theta})vt+1=βvt +∇L(θ~)
- **Parameter**: θt+1=θt−ηvt+1\theta_{t+1} = \theta_t - \eta v_{t+1}θt+1=θt−ηvt+1

**2.** AdaGrad (Adaptive Gradient Algorithm)

AdaGrad menyesuaikan learning rate untuk setiap parameter berdasarkan frekuensi pembaruan parameter. Ini memberikan learning rate yang lebih besar untuk parameter yang jarang diperbarui:

- **Update**: θt+1=θt−ηGt+ϵ∇L(θt)\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{G_t + \epsilon}} \nabla L(\theta_t)θt+1=θt−Gt+ϵη∇L(θt)
- Di sini, GtG_tGt adalah akumulasi gradien kuadrat dan ϵ\epsilonϵ adalah konstanta kecil untuk mencegah pembagian dengan nol.

**3.** RMSprop (Root Mean Square Propagation)

RMSprop adalah perbaikan dari AdaGrad yang mengatasi masalah akumulasi gradien dengan memodifikasi akumulasi gradien menjadi rata-rata eksponensial:

- **Update**: E[∇L(θt)2]=βE[∇L(θt−1)2]+(1−β)∇L(θt)2E[\nabla L(\theta_t)^2] = \beta E[\nabla L(\theta_{t-1})^2] + (1 - \beta) \nabla L(\theta_t)^2E[∇L(θt)2]=βE[∇L(θt−1)2]+(1−β)∇L(θt)2
- **Parameter**: θt+1=θt−ηE[∇L(θt)2]+ϵ∇L(θt)\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E[\nabla L(\theta_t)^2] + \epsilon}} \nabla L(\theta_t)θt+1=θt−E[∇L(θt)2]+ϵη∇L(θt)

**4.** Adam (Adaptive Moment Estimation)

Adam menggabungkan keunggulan dari Momentum dan RMSprop dengan memelihara rata-rata gradien dan kuadrat gradien:

- **Update**:
  - **Momentum (first moment estimate)**: mt+1=β1mt+(1−β1)∇L(θt)m_{t+1} = \beta_1 m_t + (1 - \beta_1) \nabla L(\theta_t)mt+1=β1mt+(1−β1)∇L(θt)
  - **Variance (second moment estimate)**: vt+1=β2vt+(1−β2)∇L(θt)2v_{t+1} = \beta_2 v_t + (1 - \beta_2) \nabla L(\theta_t)^2vt+1=β2vt+(1−β2)∇L(θt)2
  - **Bias-corrected estimates**: m^t+1=mt+11−β1t\hat{m}_{t+1} = \frac{m_{t+1}}{1 - \beta_1^t}m^t+1=1−β1tmt+1 and v^t+1=vt+11−β2t\hat{v}_{t+1} = \frac{v_{t+1}}{1 - \beta_2^t}v^t+1=1−β2tvt+1
  - **Parameter update**: θt+1=θt−ηm^t+1v^t+1+ϵ\theta_{t+1} = \theta_t - \frac{\eta \hat{m}_{t+1}}{\sqrt{\hat{v}_{t+1}} + \epsilon}θt+1=θt−v^t+1+ϵηm^t+1

    Di sini, β1\beta_1β1 dan β2\beta_2β2 adalah faktor penghalusan untuk rata-rata gradien dan kuadrat gradien, sementara ϵ\epsilonϵ adalah konstanta kecil.

**5.** Learning Rate Schedules

Learning rate schedules adalah teknik untuk mengubah learning rate selama pelatihan untuk meningkatkan konvergensi:

- **Step Decay**: Menurunkan learning rate setelah sejumlah epoch.
- **Exponential Decay**: Menurunkan learning rate secara eksponensial seiring berjalannya waktu.
- **Adaptive Learning Rates**: Menggunakan teknik seperti Cyclical Learning Rates (CLR) yang mengubah learning rate dalam rentang tertentu.

**6.** BFGS (Broyden-Fletcher-Goldfarb-Shanno)

BFGS adalah metode optimasi berbasis kuasi-Newton yang memperkirakan invers matriks Hessian (matriks kedua derivatif dari fungsi kerugian) untuk memperbarui parameter model. Ini biasanya digunakan dalam optimasi tanpa batasan.

Metode optimasi ini membantu mengatasi berbagai tantangan dalam pelatihan model pembelajaran mesin, termasuk konvergensi yang lambat, fluktuasi besar dalam fungsi kerugian, dan ketidakstabilan dalam pembaruan parameter. Memilih metode optimasi yang tepat sering kali bergantung pada sifat data, model yang digunakan, dan tujuan spesifik dari pelatihan model.

## 3. KOMPONEN KUNCI DALAM PENGOPTIMALAN MODEL

A. **Fungsi Kerugian (Loss Function)**: Mengukur seberapa baik atau buruk model melakukan prediksi. Contoh fungsi kerugian termasuk Mean Squared Error (MSE) untuk regresi dan Cross-Entropy Loss untuk klasifikasi.

B. **Fungsi Objektif**: Biasanya merupakan fungsi kerugian yang ingin diminimalkan atau dimaksimalkan selama pelatihan model.

C. **Parameter Model**: Parameter yang dipelajari selama pelatihan, seperti bobot dalam jaringan saraf.

D. **Hyperparameter**: Parameter yang tidak dipelajari dari data tetapi diatur sebelum pelatihan, seperti learning rate dan jumlah layer dalam jaringan saraf.

## 4. METODE OPTIMASI

Berikut adalah beberapa metode optimasi yang umum digunakan dalam pengoptimalan model pembelajaran mesin:

- **Gradient Descent**: Algoritma dasar untuk optimasi yang menghitung gradien fungsi kerugian dan memperbarui parameter model dalam arah yang mengurangi fungsi kerugian.
    - **Batch Gradient Descent**: Menggunakan seluruh dataset untuk menghitung gradien.
    - **Stochastic Gradient Descent (SGD)**: Menggunakan satu contoh data pada satu waktu untuk memperbarui parameter.
    - **Mini-Batch Gradient Descent**: Menggunakan subset data (mini-batch) untuk menghitung gradien dan memperbarui parameter.
- **Variasi Gradient Descent**:
    - **Momentum**: Mempercepat konvergensi dengan mempertimbangkan gradien sebelumnya dan mengakumulasi arah pergerakan.
    - **Nesterov Accelerated Gradient (NAG)**: Memperbaiki Momentum dengan menambahkan perkiraan langkah depan.
    - **AdaGrad**: Menyesuaikan learning rate untuk setiap parameter berdasarkan frekuensi pembaruan parameter.
    - **RMSprop**: Mengurangi akumulasi gradien yang telah lama untuk menyeimbangkan pembaruan parameter.
    - **Adam (Adaptive Moment Estimation)**: Menggabungkan Momentum dan RMSprop, memelihara rata-rata gradien dan kuadrat gradien.

**Regularisasi**

Regularisasi adalah teknik untuk mencegah overfitting dengan menambahkan penalti pada ukuran kompleksitas model:

- **L1 Regularization**: Menambahkan penalti proporsional terhadap nilai absolut parameter.
- **L2 Regularization**: Menambahkan penalti proporsional terhadap kuadrat nilai parameter.

- **Dropout**: Menghapus neuron secara acak selama pelatihan untuk mencegah ketergantungan yang terlalu kuat pada fitur tertentu.

**Pemilihan dan Penyelarasan Hyperparameter**

Hyperparameter mempengaruhi proses pelatihan dan kinerja akhir model. Teknik untuk memilih hyperparameter termasuk:

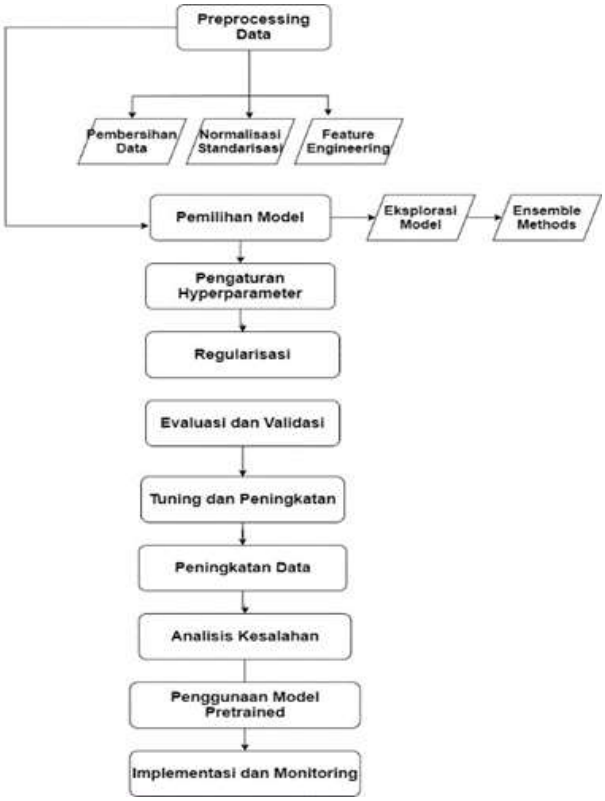
- **Grid Search**: Menguji semua kombinasi dari nilai hyperparameter yang mungkin.
- **Random Search**: Menguji kombinasi hyperparameter secara acak, sering kali lebih efisien daripada grid search.
- **Bayesian Optimization**: Menggunakan model probabilistik untuk memilih kombinasi hyperparameter yang optimal berdasarkan hasil sebelumnya.

## 5. EVALUASI MODEL

Setelah melatih model, penting untuk mengevaluasi kinerjanya menggunakan metrik yang sesuai, seperti akurasi, presisi, recall, F1-score untuk klasifikasi, atau MSE, RMSE untuk regresi. Evaluasi ini membantu memastikan bahwa model bekerja dengan baik pada data yang tidak terlihat dan tidak hanya pada data pelatihan.

**Kasus Penggunaan Dan Praktik**

Mengaplikasikan teknik pengoptimalan pada berbagai jenis model, seperti regresi linear, pohon keputusan, atau jaringan saraf, memerlukan penyesuaian dan eksperimen. Praktik terbaik melibatkan eksperimen dengan berbagai teknik optimasi, penyetelan hyperparameter, dan regularisasi untuk menemukan konfigurasi yang terbaik.




**Gambar 1: Pengoptimalan Model Pembelajaran Mesin**

Sumber : Penulis 2024

Mengoptimalkan model pembelajaran mesin adalah langkah penting dalam meningkatkan kinerja model. Berikut adalah beberapa langkah dan teknik umum yang bisa digunakan untuk mengoptimalkan model pembelajaran mesin:

**1. Preprocessing Data**

- **Pembersihan Data**: Hapus data yang tidak relevan, perbaiki kesalahan, dan tangani nilai yang hilang.
- **Normalisasi/Standarisasi**: Ubah skala fitur sehingga model dapat belajar dengan lebih baik.
- **Feature Engineering**: Ciptakan fitur baru yang dapat membantu model dalam membuat prediksi.

**2. Pemilihan Model**

- **Eksplorasi Model**: Cobalah berbagai jenis model (misalnya, regresi linier, pohon keputusan, SVM, jaringan saraf) untuk melihat mana yang paling sesuai dengan data Anda.

- **Ensemble Methods**: Gunakan metode ensemble seperti Random Forest atau Gradient Boosting untuk menggabungkan beberapa model dan meningkatkan kinerja.

**3. Pengaturan Hyperparameter**

- **Grid Search**: Uji berbagai kombinasi hyperparameter untuk menemukan set yang optimal.
- **Random Search**: Pilih secara acak kombinasi hyperparameter untuk mengeksplorasi ruang hyperparameter dengan lebih efisien.
- **Bayesian Optimization**: Gunakan metode probabilistik untuk menemukan kombinasi hyperparameter yang optimal.

**4. Regularisasi**

- **L1/L2 Regularization**: Tambahkan penalti pada model untuk mengurangi overfitting dengan mengatur ukuran koefisien fitur.
- **Dropout**: Untuk jaringan saraf, gunakan dropout untuk mengurangi overfitting dengan secara acak menonaktifkan neuron selama pelatihan.

**5. Evaluasi dan Validasi**

- **Cross-Validation**: Gunakan teknik cross-validation untuk memastikan bahwa model Anda tidak overfitting dan generalisasi dengan baik ke data yang tidak terlihat.
- **Metric Evaluation**: Pilih metrik evaluasi yang sesuai dengan masalah Anda, seperti akurasi, precision, recall, F1-score, atau AUC-ROC.

**6. Tuning dan Peningkatan**

- **Learning Rate Scheduling**: Sesuaikan laju pembelajaran selama pelatihan untuk meningkatkan konvergensi.
- **Early Stopping**: Hentikan pelatihan ketika model tidak menunjukkan peningkatan pada data validasi untuk mencegah overfitting.

**7. Peningkatan Data**

- **Augmentasi Data**: Untuk masalah seperti pengenalan gambar, tambahkan variasi pada data untuk memperluas dataset dan meningkatkan generalisasi.
- **Penambahan Data**: Kumpulkan lebih banyak data jika memungkinkan untuk melatih model dengan dataset yang lebih besar dan lebih representatif.

**8. Analisis Kesalahan**

- **Error Analysis**: Analisis kesalahan model untuk memahami pola kesalahan dan mencari cara untuk memperbaikinya.
- **Misclassification Analysis**: Identifikasi jenis data yang sering salah diklasifikasikan dan perbaiki masalah tersebut.

**9. Penggunaan Model Pretrained**

- **Transfer Learning**: Gunakan model yang telah dilatih pada dataset besar dan sesuaikan dengan data spesifik Anda.

**10. Implementasi dan Monitoring**

- **Implementasi di Produksi**: Pastikan model dapat diimplementasikan dan berfungsi dengan baik dalam lingkungan produksi.
- **Monitoring**: Pantau kinerja model secara terus-menerus dan lakukan pembaruan jika diperlukan.

Dengan mengikuti langkah-langkah ini, dapat meningkatkan kinerja model pembelajaran mesin Anda dan mencapai hasil yang lebih baik dalam berbagai tugas pembelajaran mesin.

## DAFTAR PUSTAKA

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Domingos, P. (2012). A few useful things to know about machine learning. *Communications of the ACM, 55*(10), 78-87. https://doi.org/10.1145/2347736.2347755

Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction*. Springer.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems (NeurIPS), 25*, 1097-1105.

Murphy, K. P. (2012). *Machine learning: A probabilistic perspective*. MIT Press.

Silver, D., Huang, A., Maddison, C. J., Guez, A., & Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature, 529*(7587), 484-489. https://doi.org/10.1038/nature16961