

Sistem Pakar Untuk Membantu Diagnosis Diabetes Menggunakan *Machine Learning* Dengan Algoritma Jaringan Saraf Tiruan

Satya Wirapati^{a1}, Dra. Luh Gede Astuti, M.Kom^{a2}.

^aProgram Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Udayana Badung, Bali, Indonesia

¹satyathewise@gmail.com

²lg.astuti@unud.ac.id

Abstrak

Diabetes merupakan salah satu penyakit tidak menular (PTM) yang saat ini telah menjadi ancaman serius bagi kesehatan global. Dikutip dari data World Health Organization (WHO), 70% dari total kematian di dunia disebabkan oleh penyakit tidak menular. Fakta lain yang perlu diwaspadai adalah posisi diabetes sebagai salah satu silent killer di Indonesia mungkin saja naik bila penyakit ini tidak ditangani dengan baik. Di tahun 2016, persentase kematian akibat diabetes di Indonesia mencapai 6,7% dan merupakan yang tertinggi kedua setelah Sri Lanka. Angka ini cukup tinggi karena 2 dari 3 penderita diabetes di Indonesia tidak mengetahui bahwa dirinya mengidap diabetes. Kebanyakan dari mereka baru mengakses layanan kesehatan ketika sudah dalam kondisi yang memburuk, bahkan sudah mengalami komplikasi.

Hal tersebut menjadi landasan bagi penulis untuk membuat suatu aplikasi sistem pakar menggunakan algoritma deep learning/artificial neural network yang bertujuan untuk memprediksi seseorang pasien akan terkena diabetes akut dengan besar akurasi.

Dalam Hasil dari program yang telah penulis buat, yakni penulis berhasil dalam menghitung akurasi model dengan menggunakan metode yang bernama balanced accuracy score dan menggunakan data testing untuk mengtest model. Hasilnya kita mendapatkan tingkat keakurasian sekitar 65%.

1. Pengantar

Diabetes adalah penyakit umum di dunia yang obat belum ditemukan sampai saat ini. Berdasarkan laporan Diabetes Research Center, orang yang mengalami diabetes telah bertambah dua kali lipat selama sepuluh tahun terakhir ini di seluruh dunia dan sekitar 200 juta orang telah didiagnosis dan ada sekitar enam persen peningkatan prevalensi tahunan diabetes di dunia. Karena diabetes adalah penyakit kronis yang dapat menyebabkan kerusakan yang tidak dapat diperbaiki pada anggota badan dan organ vital lainnya dalam tubuh. Penggunaan dari artificial intelligence dapat meningkatkan pendeteksian, pengontrolan penyakit, dan bisa menjadi bantuan yang besar bagi para dokter. Berdasarkan Diabetes Research Center, pendiagnosaan awal terhadap pasien yang riskan, dapat mencegah 80% komplikasi kronis tipe 2 atau menundanya. Terdapat dua jenis utama diabetes, diabetes tipe 1 yang ditandai dengan kerusakan organ pankreas dalam memproduksi insulin. Lalu ada tipe 2 ditandai dengan tidak bekerjanya insulin itu sendiri.

[1] membangun neural network dalam memprediksi konsentrasi glukosa dalam pasien diabetes tipe menggunakan analisis regresi logistik untuk menghitung rasio odds versi neuroleptik yang tidak biasa dan diagnosis diabetes di masing-masing kelompok umur, mengontrol efek populasi, dan diagnosis telah membaik prediksi diabetes menggunakan neural network fuzzy.

Pemodelan prediktif merupakan salah satu solusi untuk banyak masalah Kesehatan. Melatih neural network dapat membantu para ahli pakar dan meminimalkan kemungkinan kesalahan yang dapat terjadi dalam diagnosis diabetes. Oleh karena itu proposal penelitian ini mencoba untuk solusi yang relatif efisien dengan merancang dan menerapkan sistem untuk memprediksi diabetes pada pasien yang menggunakan jaringan saraf.

Jaringan syaraf tiruan adalah pemodelan nonlinier dari teknik komputasi cerdas yang dalam beberapa tahun terakhir menjadi kemajuan dalam komputasi dan alat pemrosesan informasi diperoleh dan posisi penting dan kemajuan dalam ilmu pengetahuan. Feedforward neural networks adalah jenis jaringan saraf tiruan yang bagus. Sedangkan backpropagation bertujuan untuk meminimalkan fungsi biaya dengan menyesuaikan bobot dan bias jaringan. Gradien fungsi biaya menentukan tingkat penyesuaian sehubungan dengan parameter seperti fungsi aktivasi, weight, bias, dll. Berikut adalah struktur jaringan saraf tiruan.

- a) Input layer: mengambil data mentah yang berasal dari data set. Dengan jumlah neuron sesuai dengan field yang ada pada dataset.
- b) Hidden layer: nilai yang berada pada hidden layer berasal dari input layer dan weight setiap hubungan antara neuron di input layer dan hidden layer. Weight dan bias ditentukan saat model machine learning di optimisasi.
- c) Output layer: fungsi unit keluaran tergantung pada aktivitas dan bobot unit tersembunyi dan hubungan antara unit tersembunyi dan keluaran.

2. Metodologi Penelitian

Persiapan Data Pasien

Mempersiapkan data set pasien yang ingin di cek, apakah pasien tersebut terkena diabetes atau tidak. Dimana data set yang diberikan berisi data kandungan glukosa, tekanan darah, ketebalan kulit, dll. Kemudian setelah data didapatkan maka akan langsung di proses menggunakan program yang sudah dibuat. Berikut adalah atribut yang diperlukan berada di dalam data set :

1. Sudah berapakahnya hamil.
2. Tingkat gula darah atau kandungan glukosa dalam darah.
3. Tekanan darah diastolic atau tekanan darah pada dinding pembuluh darah saat jantung dalam kondisi istirahat.
4. Triceps skinfold thickness merupakan salah satu jenis pengukuran skinfold thickness yang valid sebagai indikator yang menunjukkan persentase lemak tubuh seseorang.
5. Tingkat kemanjuran 2 jam serum insulin.
6. Indeks massa tubuh.
7. Diabetes pedigree function digunakan sebagai indicator Riwayat diabetes dalam keluarga.
8. Umur.
9. Apakah pasien terkena penyakit diabetes tipe 2 atau tidak.

Attribute no.	Attribute
1	Number or times pregnant (NTP)
2	Plasma glucose concentration (PGC)
3	Diastolic blood pressure (mmHg) (DBP)
4	Triceps skin-fold thickness (mm) (TSFT)
5	2-h serum insulin (mu U/mL) (H2SI)
6	Body mass index (kg/m ²) (BMI)
7	Diabetes pedigree function (DPF)
8	Age
9	Class 0 and 1 (Diagnosis of type 2 diabetes disease)

Desain Model

Untuk desain model, kami menentukannya berdasarkan parameter berikut agar model bisa menghasilkan akurasi yang tinggi dan komputasi yang serendah mungkin:

- Pertama kita harus menentukan berapa banyaknya hidden layer, dan banyaknya neuron disetiap hidden layer agar model.
- Tentukan activation-activation yang dipakai di setiap hubungan-hubungan antar layer
- Tentukan fungsi yang digunakan untuk menghitung lost cost function untuk output layer.
- Dan terakhir tentukan optimizer apa yang sebaiknya digunakan.

1) Menentukan Hidden Layer

Menurut Jeff Heaton 2008, penentuan hidden layer pada jaringan syaraf tiruan sangatlah penting, jumlah neuron hidden layer yang digunakan berpengaruh besar terhadap perubahan hasil dari pelatihan jaringan tersebut. Jika jumlah neuron hidden layer terlalu sedikit, data yang akan dihasilkan oleh jaringan sering kali tidak relevan. Namun jika neuron terlalu banyak, akan memperlambat proses pelatihan dari jaringan dan akan terjadi pelatihan yang tidak hingga. Maka dari itu, diperlukanlah pencocokan terhadap jumlah neuron hidden layer yang digunakan. Sangat banyak aturan yang digunakan untuk menentukan jumlah neuron hidden layer yang tepat untuk setiap jaringan. Beberapa aturan yang sering dipakai adalah:

1. Jumlah neuron hidden layer lebih besar dari jumlah neuron input atau jumlah neuron output.

2. Jumlah neuron hidden layer seharusnya 2/3 besar dari jumlah neuron input ditambah jumlah neuron output.
 3. Jumlah neuron hidden layer harus lebih kecil atau sama dengan dua kali jumlah input layer.
- 2) Menentukan Aktivasi
- Fungsi aktivasi berfungsi untuk menghadirkan sifat non linieritas pada input. Fungsi ini juga memiliki banyak jenis, bergantung fungsi dan kebutuhan pada perceptron, maka jenis fungsi aktivasi yang digunakan berbeda.
- Linier $(z) = z$
 - Logistic/Sigmoid $(z) = 1/1+e^{-n}$
 - Rectified Linear Unit (ReLU) $(z) = \max(0, z)$
 - Tanh $(z) = \frac{e^{2z}-e^{-2z}}{e^{2z}+e^{-2z}}$

Fungsi aktivasi yang umum digunakan adalah ReLU dan biasanya digunakan pada output dari perceptron dan beberapa jenis layer komputasi lainnya.

3) Menentukan *lost function*

Lost function merupakan metode untuk mengevaluasi seberapa baik algoritma memodelkan suatu data (Li dkk., 2019). *Lost function* memiliki kurva yang bertujuan untuk memberitahu cara mengubah parameter untuk membuat model lebih akurat. Cara kerja *lost function* adalah membandingkan hasil prediksi dari *output layer* dengan target. Ada banyak jenis *lost function* yang bisa digunakan yaitu *meansquared error loss*, *mean absolute error loss*, *multi-class cross entropy loss*, *kullback leibler divergence loss*, dll.

Pada penelitian ini, kami menggunakan metode *cross entropy*. Kami menggunakan metode ini karena metode ini memberikan penalti yang banyak. Hal ini jauh lebih baik jika dibandingkan dengan Metode MSE di mana metode ini memberikan penalti prediksi yang salah. Selain itu, metode *cross entropy* menghukum prediksi yang salah tetapi yakin serta prediksi yang benar tapi kurang percaya diri. Hal ini juga lebih baik jika dibandingkan dengan metode negatif log-likelihood yang tidak memberikan penalti berdasarkan keyakinan prediksi.

Metode *cross entropy* menghitung perbedaan antara dua distribusi probabilitas untuk serangkaian kejadian atau variabel acak yang disediakan. Ini digunakan untuk menghitung skor yang merangkum perbedaan rata-rata antara nilai yang diprediksi dan nilai sebenarnya. Untuk meningkatkan akurasi model, harus dicoba meminimalkan skor-skor lintas tropi antara 0 dan 1, dan nilai sempurna adalah 0. Berikut merupakan persamaan *cross entropy* :

$$H(p, q) = - \sum_x p(x) \log(q(x))$$

4) Menentukan *Optimizer*

Jadi untuk *Optimizer* sendiri berguna untuk meningkatkan pembelajaran dengan memperbaiki pengetahuan di jaringan, jadi neural network nantinya akan mengambil data input dan mendorongnya menjadi ansambel lapisan. Jaringan perlu mengevaluasi kinerjanya dengan fungsi kerugian. Fungsi kerugian memberi ke jaringan ide dari jalur yang perlu diambil sebelum menguasai pengetahuan. network perlu meningkatkan pengetahuannya dengan bantuan

pengoptimal. Untuk optimizer sendiri ada beberapa yang baik untuk digunakan seperti:

- SGD
- AdaDelta
- AdaGrad
- RMSProp
- Adam

Untuk penelitian ini menurut kami Adam adalah yang paling cocok. Jadi Adam adalah algoritme berbasis gradien urutan pertama dari fungsi tujuan stokastik, berdasarkan perkiraan adaptif momen tingkat rendah. Adam adalah salah satu algoritme pengoptimalan mutakhir yang digunakan oleh banyak praktisi pembelajaran mesin. Momen pertama dinormalisasi oleh momen kedua memberikan arah pembaruan. Jadi menurut kami jika ingin melatih neural network dalam waktu yang lebih singkat dan lebih efisien daripada Adam adalah optimizer yang terbaik. Untuk sparse data, lebih baik menggunakan pengoptimal dengan kecepatan pembelajaran dinamis. Jika, ingin menggunakan gradient descent algorithm dari pada penurunan min-batch gradient descent adalah pilihan terbaik.

3. Hasil dan Pembahasan

Cell 1

Pada cell 1 terdapat syntax yang berisikan library-library yang dibutuhkan sedang diimport. Setelah itu, file csv yang ingin dijadikan dataset harus diubah terlebih dahulu ke dalam pandas data frame menggunakan metode pandas, yakni `read_csv` ke dalam variabel "dataset" dan di dalam argumennya yang pertama path file csv dimasukkan dan kedua menentukan delimiter yang ada di file csv. Kemudian pisahkan field/independent variable dan output data menggunakan method `iloc`. Dan terakhir data dibagi menjadi data training dan data test dengan perbandingan 4:1 secara refektif dengan menggunakan method `train_test_split`.

```
from numpy import loadtxt
import pandas as pd
from keras.models import Sequential
from keras.layers import Dense
from sklearn.model_selection import train_test_split
import numpy as np
from sklearn.metrics import balanced_accuracy_score

dataset = pd.read_csv("Book1.csv", delimiter=",")
X = dataset.iloc[:, 0:8]
y = dataset.iloc[:,8]

#bagi data menjadi data latih dan data test dengan ratio 8:2
x_train,x_test,y_train,y_test=train_test_split(X, y, test_size=0.2)
```

Cell 2

Kemudian model neural networknya dibuat dengan cara membuat objek `sequential()` ke dalam variabel "model" yang diimport dari library keras. Lalu dengan metode `add`, ditambahkan layer pertama dan kedua pada pemanggilan metode `add` yang pertama, di dalam metode `add` pertama dimasukkan argument yang pertama, yaitu banyaknya neuron pada hidden layer kedua, yaitu 12, lalu argumen kedua banyaknya neuron pada input layer yaitu sesuai dengan independent variable pada data set dan argument ketiga, yaitu menentukan aktivasi di antar layer tersebut, dan di sini kami memilih relu activation. Kami memilih relu karena ReLu (Rectified Linear Unit) memiliki fungsi turunan dan memungkinkan backpropagation sekaligus membuatnya

efisien secara komputasi. Neuron hanya akan dinonaktifkan jika output dari transformasi linier bernilai kurang dari 0. Kemudian pada pemanggilan metode `add` yang ke dua, di argument pertama dimasukkan banyaknya neuron pada hidden layer ketiga, yaitu 8, dan aktivasinya sama antara hidden layer pertama dan input layer, yaitu ReLu activation. Dan yang terakhir adalah output layer, outputnya ditentukan 1 dan menggunakan activation sigmoid. Sigmoid activation function mengambil nilai nyata apa pun sebagai nilai input dan output dalam kisaran 0 sampai 1. Semakin besar nilai inputnya (semakin positif), maka nilai outputnya akan semakin dekat dengan 1, sedangkan semakin kecil nilai inputnya (semakin negatif), maka semakin dekat nilai outputnya dengan 0. Alasan kita memilih sigmoid adalah karena hasil output yang besar, akan menghasilkan komputasi dan memakan penyimpanan yang besar juga jadi kami meminilisirnya dengan menggunakan sigmoid.

```
model = Sequential()  
model.add(Dense(12, input_dim=8, activation="relu"))  
model.add(Dense(8, activation="relu"))  
model.add(Dense(1, activa
```

Cell 3

Pemodelan terakhir pada model nn ini adalah menentukan fungsi loss cost dan fungsi optimizer, untuk lost cost digunakan binary cross entropy dan untuk optimizer digunakan adam. Binary cross entropy digunakan untuk klasifikasi biner yang mana nilai target berada pada set {0,1}.Adam digunakan untuk optimizer karena efisien secara komputasi/perhitungan,sesuai dalam hal data atau parameter yang memiliki masalah yang besar, dan mudah untuk diterapkan. Dan terakhir mem-fit kan, yakni menentukan banyaknya epochs=150 dan banyaknya sampel data di setiap batch `batch_size=32`.

```
model.compile(loss="binary_crossentropy", optimizer="adam",  
metrics=["accuracy"] )  
model.fit(x_train, y_train, epochs=150, batch_size=32)  
_, accuracy = model.evaluate(x_train, y_train)
```

Cell 6

Akurasi model dihitung dengan menggunakan metode balanced accuracy score dan menggunakan data testing untuk mengetest model. Hasilnya didapatkan tingkat ke-akurasian sekitar 65%.

```
from sklearn.metrics import accuracy_score  
predictions = (model.predict(x_test) > 0.5).astype(int)  
balanced_accuracy_score(y_test, p
```

4. Kesimpulan

Dari Hasil program yang telah kami buat yang sudah dijelaskan sebelumnya, Hal pertama yang dapat dilakukan dalam metode penelitian yaitu mempersiapkan data set pasien yang kemudian akan langsung di proses menggunakan program yang sudah dibuat. Dalam desain model kami menentukan berdasarkan beberapa parameter yang bertujuan untuk menghasilkan akurasi yang tinggi dan komputasi yang serendah mungkin. Parameter-parameter tersebut antara lain Menentukan : Hidden Layer, Aktivasi, Lost Function, serta mentukan Optimizer. Pada Optimizer sendiri kami menggunakan algoritme berbasis gradien urutan pertama dari fungsi tujuan stokastik, berdasarkan perkiraan adaptif momen tingkat rendah yang bisa disebut dengan Adam. Yang dimana Adam adalah salah satu algoritme pengoptimalan mutakhir yang digunakan oleh banyak praktisi pembelajaran mesin.

Daftar Pustaka

- [1] S. C. B. a. R. Pappada, " Development of a neural network for prediction of glucose concentration in type 1," *Journal of Diabetes Science and Technology*, pp. 792-801,2008.
- [2] P. RAHIMLOO and A. JAFARIAN, "Prediction of Diabetes by Using Artificial Neural Network,," *Bulletin de la Société Royale des Sciences de Liège*, pp. 1148 - 1164, 2016.
- [3] E. L. J. L. C. X. J. T. T. Y. M. e. a. Pei, "" Effects of lipids and lipoproteins on," *JDiabetes*, p. 559–64, 2014.
- [4] J. O. Orukwo and L. G. Kabari, "Diagnosing Diabetes Using Artificial Neural Networks," p. 5, 2020.
- [5] D. Nielsen, L. Krych and K. Buschard, "Beyond Genetics Influence of dietary factorsand gut," *FEBS Lett*, p. 4234–43, 2014.
- [6] d. F. R. Makarim, "Diabetes," [Online]. Available: <https://www.halodoc.com/kesehatan/diabetes>.
- [7] D. Godoy, "Understanding binary cross-entropy / log loss: a visual explanation," 22 November 2018. [Online]. Available: <https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a>.
- [8] P. Baheti, "12 Types of Neural Network Activation Functions: How to Choose?," 26May 2022. [Online]. Available: <https://www.v7labs.com/blog/neural-networks-activation-functions>.

This page is intentionally left blank