BAB 2
TINJAUAN PUSTAKA

2.1. Deep Learning

Deep learning merupakan salah satu bidang dari machine learning yang memanfaatkan jaringan syaraf tiruan untuk implementasi permasalahan dengan dataset yang besar. Teknik deep learning memberikan arsitektur yang sangat kuat untuk supervised learning[7]. Dengan menambahkan lebih banyak lapisan maka model pembelajaran tersebut bisa mewakili data citra berlabel dengan lebih baik. Pada machine learning terdapat teknik untuk menggunakan ekstraksi fitur dari data pelatihan dan algoritma pembelajaran khusus untuk mengklasifikasi citra maupun untuk mengenali suara[7]. Namun, metode ini masih memiliki beberapa kekurangan baik dalam hal kecepatan dan akurasi. Aplikasi konsep jaringan syaraf tiruan yang dalam (banyak lapisan) dapat ditangguhkan pada algoritma Machine learning yang sudah ada sehingga komputer sekarang bisa belajar dengan kecepatan, akurasi, dan skala yang besar [7].

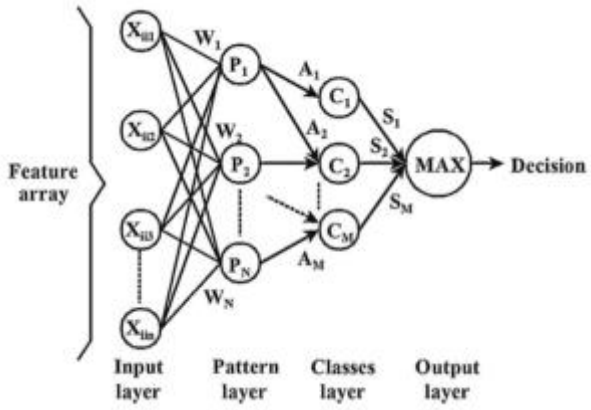
Prinsip ini terus berkembang hingga Deep learning semakin sering digunakan pada komunitas riset dan industri untuk membantu memecahkan banyak masalah data besar seperti computer vision, speech recognition, dan Natural Language Processing[7]. Feature Engineering adalah salah satu fitur utama dari Deep learning untuk mengekstrak pola yang berguna dari data yang akan memudahkan model untuk membedakan kelas. Feature Engineering juga merupakan teknik yang paling penting untuk mencapai hasil yang baik pada tugas prediksi. Namun, sulit untuk dipelajari dan dikuasai karena kumpulan data dan jenis data yang berbeda memerlukan pendekatan teknik yang berbeda juga.

Algoritma yang digunakan pada Feature Engineering dapat menemukan pola umum yang penting untuk membedakan antara kelas Dalam Deep learning, metode PNN atau Probabilistic neural network sangatlah bagus dalam menemukan fitur yang baik pada citra ke lapisan berikutnya untuk membentuk hipotesis nonlinier yang dapat meningkatkan kekompleksitasan sebuah model. 21 Model yang

kompleks tentunya akan membutuhkan waktu pelatihan yang lama sehingga di dunia Deep learning pengunaan GPU sudah sangatlah.

2.2 Probabilistic Neural Network

PNN pada dasarnya adalah jaringan syaraf tiruan backpropagation dengan perbedaan pada fungsi aktivasinya yang menggunakan data statistik[2]. PNN termasuk ke dalam supervised learning, yaitu membutuhkan data latih dan kelas target untuk proses pembuatan model. Arsitektur PNN disajikan pada Gambar 2.1.



Gambar 2. 1 Struktur PNN

arsitektur PNN terdiri atas 4 lapisan, lapisan tersebut adalah:

1. Lapisan Masukan (Input Layer) Lapisan ini merupakan masukan yang berupa vektor x dengan jumlah elemen sebanyak n. Vektor tersebut akan diklasifikasikan pada salah satu kelas dari total keseluruhan M kelas. Seluruh elemen pada vektor x terhubung penuh ke lapisan berikutnya.
2. Lapisan Pola (Pattern Layer) Lapisan kedua yaitu lapisan pola. Perkalian titik (dot product) antara vektor input x dan vektor bobot pelatihan dilakukan. Kemudian hasil perkalian tersebut dimasukan ke dalam fungsi radial basis Gausian, yaitu radbas (n) = exp (-n). Lapisan pola berfungsi untuk menghitung jarak antara vektor input dan vektor bobot pelatihan yang dipresentasikan oleh neuron. Persamaan yang digunakan pada lapisan pola adalah sebagai berikut:

$$f(x) = \exp\left(-\frac{(x - x_{ij})^T (x - x_{ij})}{2\,\sigma^2}\right)$$

3. Lapisan Penjumlahan (Summation Layer) Lapisan penjumlahan menerima masukan dari lapisan pola yang terkait dengan kelas yang ada. Lapisan ini memiliki satu neuron untuk setiap kelas. Pada neuron-neuron tersebut ditampung hasil penjumlahan dari setiap kelas dari lapisan pola. Persamaan yang digunakan pada lapisan ini yaitu:

$$p(x) = \frac{1}{(2\pi)^{k/2}\,\sigma^k t} - \frac{1}{N^t}\sum_{i=1}^{t} f(x)$$

4. Lapisan Keluaran (Output Layer) Lapisan terakhir pada PNN yaitu lapisan keluaran. Pada lapisan ini dihasilkan sebuah vektor dengan panjang M elemen, M menyatakan banyaknya kelas.

2.3 Unified Modelling Language

Dalam membangun aplikasi berorientasi objek, salah satu model perancangan yang digunakan adalah *Unified Modelling Language* (UML). UML adalah sebuah standar bahasa berdasarkan grafik/gambar yang berguna untuk memvisualisasi, menspesifikasikan, membangun, dan mendokumentasikan sebuah pengembangan sistem perangkat lunak *software* berbasis OO *(Object-Oriented*) [23]. Dalam UML terdapat berbagai macam diagram yang digunakan untuk memodelkan aplikasi berorientasi objek, antara lain: *Use Case, Activity Diagram, Sequence Diagram, dan Class Diagram*. UML digunakan agar rancangan sistem lebih mudah dipahami oleh banyak pihak yang terlibat dalam proses pengembangannya.

2.3.1 Use Case Diagram

Use case diagram digunakan untuk memodelkan proses bisnis berdasarkan perspektif pengguna sistem. *Use case diagram* terdiri dari diagram untuk *use case* dan *actor*, di mana *actor* merupakan representasi dari orang yang akan mengoperasikan orang yang berinteraksi dengan sistem, namun bisa juga sebuah sistem yang lain [23].

2.3.2 Activity Diagram

Activity diagram menggambarkan berbagai aliran aktivitas yang ada pada sistem yang dirancang, mulai dari awal dari masing-masing alir berawal, *decision* atau pilihan yang mungkin terjadi, hingga bagaimana aliran tersebut berakhir. Keunggulan dari *activity diagram* adalah lebih mudah dimengerti jika dibandingkan skenario [23].

2.3.3 Class Diagram

Class diagram menggambarkan struktur statis dari kelas dalam sistem anda dan menggambarkan atribut, operasi dan hubungan antara kelas [23]. *Class diagram* berfungsi untuk memvisualisasikan struktur kelas-kelas dari suatu sistem. Suatu Class memiliki tiga bagian pokok, yaitu nama (dan *stereotype*), atribut, dan metoda yang dapat dilakukannya. Suatu *class* dapat memiliki hubungan antara satu sama lain melalui berbagai cara, antara lain: *associated* (terhubung satu sama lain), *dependent* (satu *class* bergantung atau menggunakan class yang lain), *specialized* (satu *class* merupakan spesialisasi dari *class* lainnya), atau *package* (grup bersama sebagai satu unit). Di dalam sebuah sistem biasanya terdapat beberapa *class diagram*.

2.3.4 Sequence Diagram

Sequence diagram menggambarkan interaksi objek dalam sistem yang disusun berdasarkan urutan waktu. *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Objek yang berkaitan dengan proses berjalannya operasi diurutkan dari kiri ke kanan sesuai dengan waktu terjadinya dalam pesan yang terurut [23].

2.4 Python

Python adalah salah satu bahasa pemrograman komputer yang populer dan sering digunakan terutama dalam melakukan perhitungan. Bahasa pemrograman ini dapat dijalankan hampir di semua platform, seperti *Linux*, *Windows*, dan *Machintos*. Pada bahasa pemrograman *Python*, deklarasi suatu variabel dapat dilakukan secara langsung tanpa menyebutkan tipe datanya.

Bahasa pemrograman *Python* diciptakan oleh Guido van Rossum, dan diperkenalkan pertama kali pada Centrum Wiskunde & Informatica (CWI) di Belanda pada awal tahun 1990-an. Bahasa pemrograman yang terinspirasi dari bahasa pemrograman ABC ini pertama kali dikembangkan pada awal tahun 1990 oleh Guido van Rossum di Stiching Mathematisch Centrum. Pada tahun 1995, pengembangan bahasa pemrograman *Python* dilakukan di Corporation for National Research Initiatives. Saat ini *Python Software Foundation* bertugas sebagai pengembang *Python* [23].

2.5 Citra Digital

Citra adalah suatu representasi (gambaran), kemiripan, atau imitasi dari suatu objek. Citra terbagi 2 yaitu ada citra yang bersifat analog dan ada citra yang bersifat digital. Citra analog adalah citra yang bersifat kontinu seperti gambar pada monitor televisi, foto sinar X, hasil CT Scan dll. Sedangkan pada citra digital adalah citra yang dapat diolah oleh komputer(T,Sutoyo et al. 2009: 9). Citra atau gambar yang didefinisikan sebagai sebuah fungsi dua dimensi, f(x,y), dimana x dan y adalah koordinat bidang datar, dan harga fungsi f disetiap pasangan koordinat (x,y) disebut intensitas atau level keabuan (grey level) dari gambar dititik itu. Jika x,y dan f semuanya berhingga (finite) dan nilainya diskrit maka gambarnya disebut citra digital [15].

2.6 Citra Grayscale

Citra grayscale, merupakan citra yang nilai piksel-nya merepresentasikan derajat keabuan atau intensitas warna putih. Nilai intensitas paling rendah adalah merepresentasikan warna hitam dan nilai intensitas paling tinggi merepresentasikan warna putih. Banyaknya warna pada jenis citra grayscale bergantung pada jumlah bit yang disediakan oleh memori untuk menampung. Citra 2 bit mewakili 4 warna, citra 3 bit mewakili 8 warna, dan seterusnya sampai 8 bit. Untuk mengubah citra berwarna menjadi citra grayscale dilakukan konversi dengan menggunakan persamaan dimana: R = Unsur warna merah pada basis warna RGB G = Unsur warna hijau pada basis warna RGB B = Unsur warna biru pada basis warna RGB

2.7. Android

Android adalah sebuah sistem operasi berbasis linux yang dirancang untuk perangkat mobile layar sentuh seperti telepon pintar dan komputer tablet. Android bersifat terbuka atau open source yang memungkinkan pengembang menciptakan aplikasinya sendiri. Pada awalnya Android dikembangan oleh 16 Android Inc., kemudian Google membelinya pada tahun 2005. Sistem operasi ini dirilis pada tahun 2007 bersamaan dengan didirikannya Open Handset Alliance. Ponsel Android pertama dijual pada tahun 2008. Terdapat dua jenis distributor sistem operasi Android yaitu pertama yang mendapat dukungan penuh dari Google atau Google Mail Services (GMS) dan yang kedua adalah benar-benar bebas distribusinya tanpa dukungan Google atau dikenal sebagai Open Handset Distribution (OHD)