BAB 2

TINJAUAN PUSTAKAN DAN DASAR TEORI

2.1 Tinjauan Pustaka

Sebagai keperluan melakukan perbandingan tinjauan pustaka dengan penelitian, maka dibuat tabel sebagai berikut:

Tabel 2.1 Tinjauan Pustaka

Penulis	Objek Penelitian	Metode Penelitian	Hasil Penelitian
Han et al (2015)	ImageNet dataset	Pruning, Trained Quantization, Huffman code	Mengurangi ukuran dari VGG-16 sebanyak 49 kali dari 522 MB menjadi 11.3 MB tanpa mengurangi akurasi model
Gong et al (2014)	1000 kategori dari ImageNet dataset	Vector quantization	Mengurangi ukuran CNN sekitar 16-24 kali dengan hanya kehilangan 1% dari akurasi

LeCun et al (1990)	*Neural network*	Menggunakan informasi turunan kedua untuk membuat keseimbangan antara kompleksitas dan error	*Mengkonfirmasi kegunaan metode pada aplikasi dunia nyata*
Polino et al (2018)	*Deep Neural Network (DNNs)*	Quantized distillation, differentiable quantization	*memungkinkan DNNs untuk lingkungan dengan sumber dayaterbatas untuk meningkatkan arsitektur dan peningkatan akurasi yang dikembangkan pada perangkat yang lebih kuat.*
Schmidhuber, J. (2015).	*Neural network*	review	*Ulasan tentang supervised learning, unsupervised learning, reinforcement learning, dan evolutional computation*
LeCun et al (2015)	*Deep neural*	Deep convolutional net	*Mengkonfirmasi kegunaan metode*

	network (DNNs)		
Krizhevsky et al (2012)	*ImageNet datasets*	Deep convolutional neural network	*AlexNet*
Karen Simonyan, & Andrew Zisserman. (2014).	*ImageNet datasets*	Deep convolutional neural network	*VGGNet*
Anang Suwasto (2020)	*Cat vs Dogs datasets*	Pruning dan Quantization	*Dalam proses*

Han et al (2015). Melakukan penelitian untuk melakukan kompresi *deep neural network* (DNNs) dengan memperkenalkan suatu metode yang bernama "*deep compression*", berupa tiga tahapan yaitu: *pruning, trained quantization, huffman coding* yang bekerja bersama untuk mengurangi memori yang di butuhkan oleh *neural network* sebesar 35 hingga 49 kali tanpa mempengarusi akurasi dari model tersebut.

Gong et al (2014). Melakukan penelitian untuk melakukan kompresi *deep convolutional neural network* (CNN) menggunakan metode *vector quantization* dengan menggunakan *k-means clustering* pada bobot jaringan, untuk 1000 kategori pada ImageNet dataset berhasil menghasilkan 26 hingga 24 kali tingkat kompresi dengan hanya kehilangan 1% akurasi menggunakan *state-of-the-art* CNN.

LeCun et al (1990). Melakukan penelitian “Optimal brain damage” dalam penelitiannya peneliti menemukan dengan menghapus bobot yang tidak penting dari suatu jaringan saraf tiruan, beberapa peningkatan dapat di harapkan: generalisasi yang lebih baik, lebih sedikit contoh pelatihan yang di butuhkan dan juga kecepatan dalam mempelajari contoh.

Polino et al (2018). Melakukan penelitian untuk kompresi *deep neural network* (DNNs) dan memperkenalkan metode *quantized distillation & leverages distillation* dan yang kedua adalah *differentiable quantization*, peneliti memvalidasi metodenya dengan melakukan eksperimen pada *convolutional* dan *recurrent* arsitektur.

Schmidhuber, J. (2015). Melakukan peninjauan terhadap *supervised & unsupervised learning* (juga merekapitulasi sejarah dari *backpropagation*), *reinforcement learning*, evolusi dari komputasi, dan pencarian secara tidak langsung untuk program pendek yang di gunakan untuk *encoding* jaringan yang dalam dan besar.

LeCun et al (2015). Melakukan peninjauan terhadap *deep learning* di mana metode ini di anggap telah berhasil dalam pemrosesan gambar, video, percakapan dan audio, dan juga sangat berguna pada domain yang lain seperti penemuan obat dan genom.

Krizhevsky et al (2012). Melakukan penelitian menggunakan *deep convolutional neural network* untuk klasifikasi 1.2 juta gambar dengan resolusi tinggi, *neural network* yang di hasilkan memiliki 60 juta parameter dan 650,000 *neurons*, yang di dalamnya ada lima *convolutional layer*, beberapa di antaranya di

ikuti dengan *max-pooling layer* dan tiga *fully-connected layer* dengan 1000 *softmax*.

Karen Simonyan, & Andrew Zisserman. (2014). Melakukan penelitian yang berfokus pada kedalaman *convolutional network* dan pengaruhnya pada akurasi, jaringan yang di hasilkan di beri nama dengan VGGNet yang memiliki kedalaman 16 dan 19 layer bobot.

2.2 Dasar Teori

2.2.1 Big Data

Big Data umumnya mengacu pada data yang melebihi kapasitas penyimpanan, pemrosesan, dan komputasi tipikal dari basis data konvensional dan teknik analisis data. Sebagai sumber daya, Big Data membutuhkan alat dan metode yang dapat diterapkan untuk menganalisis dan mengekstraksi pola dari data skala besar. Munculnya Big Data telah disebabkan oleh peningkatan kemampuan penyimpanan data, peningkatan daya pemrosesan komputasi, dan ketersediaan volume data yang meningkat, yang memberi organisasi lebih banyak data daripada yang harus diproses sumber daya dan teknologi komputasi. Selain volume data yang besar dan jelas, Big Data juga dikaitkan dengan kompleksitas spesifik lainnya, yang sering disebut sebagai empat V: Volume, Variety, Velocity, dan Veracity (Najafabadi et al., 2015).

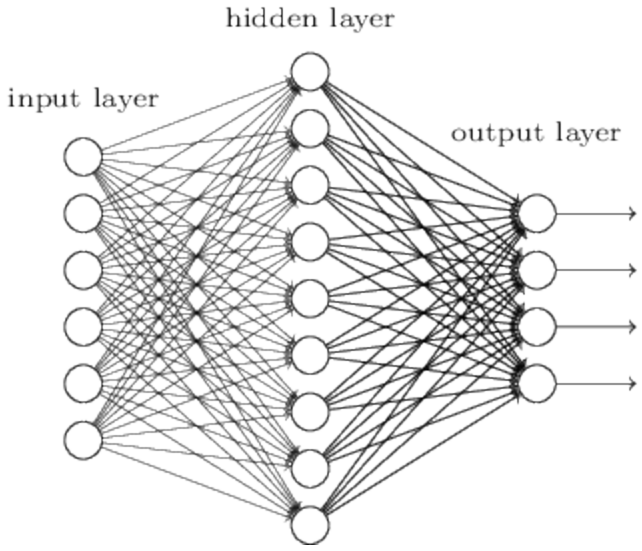
Algoritma Deep Learning mengekstraksi abstraksi tingkat tinggi dan kompleks sebagai representasi data melalui proses pembelajaran hierarkis. Abstraksi kompleks dipelajari pada level tertentu berdasarkan abstraksi yang relatif lebih sederhana yang diformulasikan pada level sebelumnya dalam hierarki.

Manfaat utama Deep Learning adalah analisis dan pembelajaran sejumlah besar data tanpa pengawasan, menjadikannya alat yang berharga untuk Big Data Analytics di mana data mentah sebagian besar tidak berlabel dan tidak dikategorikan (Najafabadi et al., 2015).

2.2.2 Deep Learning

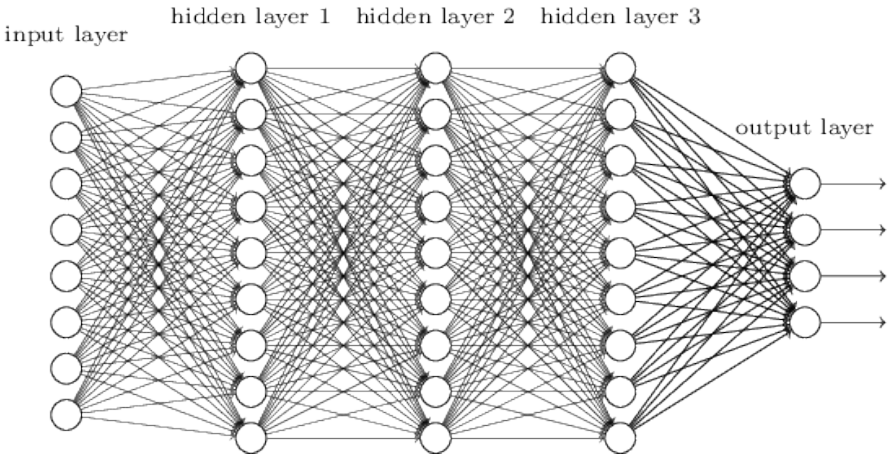
Konsep utama dalam algoritma *Deep Learning* adalah mengotomatisasi ekstraksi representasi (abstraksi) dari data (Y. Bengio et al., 2013). Algoritma *Deep Learning* menggunakan sejumlah besar data untuk secara otomatis mengestraksi representasi kompleks dari data tersebut. Algoritma ini dimotivasi oleh salah satu bidang kecerdasan buatan di mana tujuannya adalah untuk meniru kemampuan otak manusia dalam mengamati, menganalisis, belajar dan membuat keputusan, terutama pada masalah yang kompleks.

Algoritma yang berusaha meniru kemampuan otak manusia biasa di sebut dengan *Neural Network*. *Neural Network* dapat berupa *recurrent* atau *feedforward*, *feedforward* tidak memiliki perulangan di dalamnya dan dapat di susun dalam lapisan. *Neural Network* yang mempunyai dua atau lebih hidden layer bisa di kategorikan ke dalam *Deep Neural Network.*



Gambar 2.1 “Non deep” feedforward neural network

Gambar 2.1 merupakan feedforward neural network dimana jaringan terdiri dari satu input layer, satu hidden layer dan output layer. Algoritma Deep Learning sebenarnya adalah adalah arsitektur yang terdiri dari lapisan-lapisan yang berurutan. Setiap lapisan menerapkan transformasi nonlinear pada inputnya dan memberikan representasi dalam outputnya. Tujuannya adalah untuk mempelajari representasi data yang rumit dan abstrak secara hierarkis dengan melewatkan data melalui berbagai lapisan transformasi. Data sensorik (misalnya piksel dalam suatu gambar) diumpankan ke lapisan pertama. Akibatnya output dari setiap lapisan disediakan sebagai input ke lapisan berikutnya (Najafabadi et al., 2015).

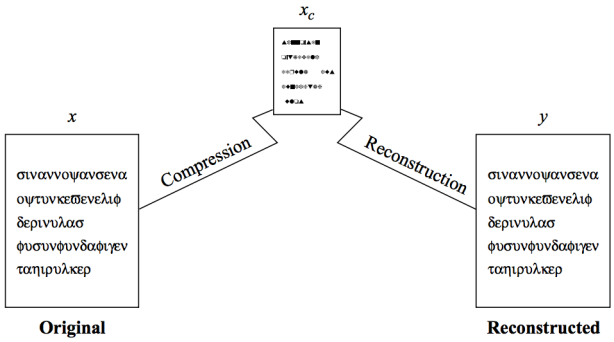


Gambar 2.2 Deep Neural Network

Gambar 2.2 merupakan deep neural netork dimana jaringan memiliki tiga hidden layers.

2.2.3 Kompresi

Kompresi data adalah seni atau sains untuk mewakili data dalam bentuk ringkas. Representasi yang lebih ringkas di buat dengan mengidentifikasi dan menggunakan struktur yang ada di dalam data. Data bisa berupa file teks, suara, gelombang, gambar atau urutan angka yang di hasilkan dari proses lain. Alasan dibutuhkanya kompresi data adalah informasi yang di hasilkan dalam bentuk digital semakin hari semakin banyak dalam bentuk angka yang di representasikan oleh byte data. Jumlah byte yang diperlukan untuk merepresentasikan data multimedia bisa sangat besar. Misalnya, untuk secara digital mewakili 1 detik video tanpa kompresi (menggunakan format CCIR 601), dibutuhkan lebih dari 20 megabytes atau 160 megabits (Sayood, Khalid., 2006).



Gambar 2.3 Teknik kompresi

(Sumber: Sayood, Khalid., 2006)

Gambar 2.3 merupakan algoritma kompresi yang mengambil input X dan menghasilkan representasi Xc yang membutuhkan bit lebih sedikit, dan ada algoritma rekonstruksi yang beroperasi pada representasi terkompresi Xc untuk menghasilkan rekonstruksi Y.

Berdasarkan persyaratan rekonstruksi, skema kompresi data dapat dibagi ke dalam dua kelas besar: skema kompresi *lossless*, dalam bentuk yang masih identik, dan skema kompresi *lossy*, yang umumnya memberikan kompresi jauh lebih tinggi daripada *lossless* kompresi tetapi memungkinkan untuk berbeda bentuk (Sayood, Khalid., 2006).

A. Kompresi Lossless

Teknik kompresi *lossless* tidak melibatkan kehilangan informasi. Jika data telah dikompresi tanpa kehilangan, data asli dapat dipulihkan dengan tepat dari data terkompresi. Kompresi *lossless* umumnya digunakan untuk aplikasi yang tidak dapat ditoleransi perbedaan antara data asli dan data yang direkonstruksi.

B. Kompresi Lossy

Teknik kompresi *lossy* melibatkan beberapa kehilangan informasi, dan data yang telah dikompresi menggunakan teknik *lossy* umumnya tidak dapat dipulihkan atau direkonstruksi dengan tepat. Sebagai imbalan untuk menerima distorsi ini dalam rekonstruksi, kita umumnya dapat memperoleh banyak rasio kompresi yang lebih tinggi daripada yang dimungkinkan dengan kompresi *lossless*.

2.2.4 Pruning

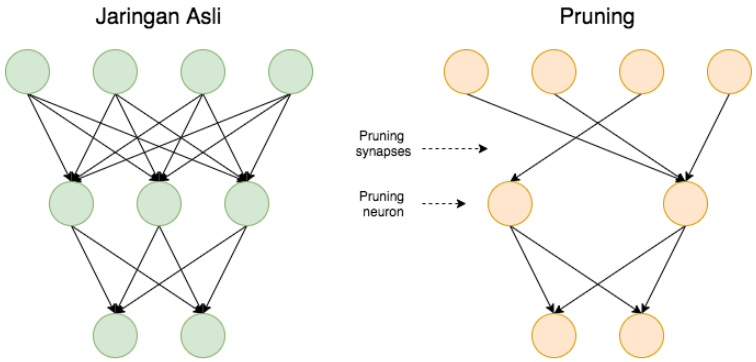
Metode umum untuk menginduksi *sparsity* dalam bobot dan aktivasi *neural network* disebut *pruning*. *Pruning* adalah penerapan kriteria biner untuk menentukan bobot yang harus di pangkas, bobot yang cocok untuk kriteria pemangkasan di beri nilai nol. Elemen yang di pangkas dari model di beri nilai nol dan di pastikan bahwa elemen tersebut tidak ambil bagian pada proses *backpropagation.*

Sparsity adalah ukuran berapa banyak elemen dalam tensor yang sama dengan nol, relatif terhadap ukuran tensor. Fungsi $l_0 - "norm"$ dapat mengukur seberapa banyak elemen nol pada suatu tensor (Zmora et al., 2019).

$$\|X\|_0 = |X_1|^0 + |X_2|^0 + \cdots + |X_n|^0$$

Weight pruning atau model pruning adalah suatu metode untuk meningkatkan *sparsity* (jumlah nol dalam sebuah tensor) dari bobot jaringan. Secara umum “parameter” merujuk pada tensor bobot dan bias pada suatu model. Pruning membutuhkan kriteria untuk memilih elemen mana yang akan di pangkas, hal ini di sebut *pruning criteria*. Cara yang paling umum adalah nilai absolute dari

elemen di bandingkan dengan nilai *treshold* jika di bawah nilai *treshold* akan di pangkas menjadi nol. Bagian yang paling sulit untuk menginduksi *sparsity* melalui *pruning* adalah menentukan *treshold* atau level *sparsity*. *Sensitivity analysis* adalah sebuah metode untuk mengurutkan tensor berdasarkan sensitivitasnya untuk di pangkas (Zmora et al., 2019).



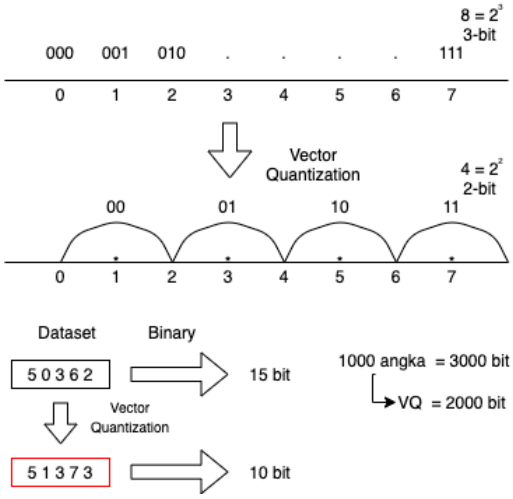
Gambar 2.4 Ilustrasi pruning

Gambar 2.4 merupakan ilustrasi dari pruning dimana pada jaringan asli semua parameter terkoneksi dari satu neuron ke neuron lain pada layer selanjutnya, sedangkan pada jaringan yang sudah di pangkas tidak semua parameter terkoneksi.

2.2.5 Quantization

Quantization mengacu pada proses mengurangi jumlah bit yang mewakili angka. Dalam konteks *deep learning*, format numerik dominan yang digunakan untuk penelitian dan untuk penyebaran sejauh ini adalah floating point 32-bit, atau FP32. Namun, keinginan untuk mengurangi bandwidth dan menghitung kebutuhan model pembelajaran dalam telah mendorong penelitian untuk menggunakan format numerik presisi rendah. Telah ditunjukkan secara luas bahwa bobot dan aktivasi dapat direpresentasikan menggunakan bilangan bulat 8-bit (atau INT8) tanpa

menimbulkan kehilangan akurasi secara signifikan. Penggunaan bit-width yang lebih rendah, seperti 4/2/1-bit, adalah bidang penelitian aktif yang juga menunjukkan kemajuan besar (Zmora et al., 2019).



Gambar 2.5 Ilustrasi Quantization

Gambar 2.5 merupakan ilustrasi quantization, misalkan kita mempunyai data dengan range 0-8 maka kita memerlukan 3bit untuk merepresentasikan satu angka, setelah *vector quantization* range data di klaster menjadi 4 dan data di ubah menjadi *centroid* nya sehingga jumlah bit yang di perlukan untuk merepresentasikan angka berkurang menjadi 2bit.

2.2.6 Akurasi

Akurasi adalah salah satu metrik untuk mengevaluasi model klasifikasi. Secara informal, akurasi adalah fraksi prediksi model kita yang benar. Secara formal, akurasi memiliki definisi berikut:

$$\text{Akurasi} = \frac{\text{Jumlah prediksi benar}}{\text{Jumlah total prediksi}}$$