**TUGAS DEEP LEARNING**



Dosen Pengampu :

Daniel Arsa, S.Kom., M.S.I

Disusun Oleh :

Nama : M. Febrin Dafian

NIM : F1E119035

Ruang : R001

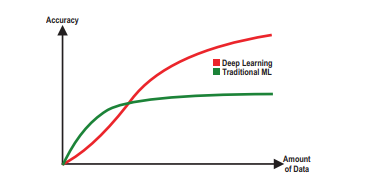
**PROGRAM STUDI SISTEM INFORMASI**

**FAKULTAS SAINS DAN TEKNOLOGI**

**UNIVERSITAS JAMBI**

**Deep Learning and Neural Networks Overview**

Deep Learning merupakan metode pilihan bagi machine learning yang lebih optimal, Perbandingan kinerja antara pembelajaran mendalam dan algoritma pembelajaran mesin tradisional ditampilkan sebagai berikut :



Deep learning adalah subbidang machine learing yang meniru kemampuan pemrosesan data dan pembuatan pola otak manusia untuk pengambilan keputusan otomatis. Deep Learning dimungkinkan berkat jaringan saraf tiruan. Jaringan syaraf tiruan adalah struktur jaringan yang mensimulasikan neuron-neuron pada otak manusia sehingga dapat terjadi deep learning.

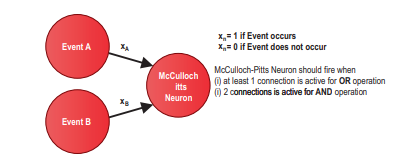
**Timeline of Neural Networks and Deep Learning Studies**

Sejarah jaringan syaraf tiruan dan deep learnig dimulai dari tahun 1943 :

* **Pengembangan neuron buatan**, tahun 1943 warren McCulloch menerbitkan makalah “A Logical Calculus of the Ideas Immanent in Nervous Activity,” mengenai moel matematika neuron biologis.
* **Backpropagation**, Istilah backpropagation diciptakan pada tahun 1986 oleh Rumelhart, Hinton, dan Williams, dan para peneliti ini telah mempopulerkan penggunaannya dalam jaringan saraf tiruan.
* **Pelatihan dan Komputerisasi**, Pada tahun 1971, Ivakhnenko menciptakan jaringan saraf delapan lapis, yang dianggap sebagai Deep Learning Network karena struktur multilayernya.
* **AI Winter**, Marvin Minsky dan Seymour Papert menulis buku Perceptrons di mana ia dengan keras menyerang karya Frank Rosenblatt, Perceptron. Buku ini menyebabkan kerusakan parah pada dana proyek AI, yang memicu musim dingin AI yang berlangsung dari tahun 1974 hingga 1980.
* **Convolutional Neural Networks**,Pada tahun 1980, Kunihiko Fukushima memperkenalkan neocognitron, convolutional neural networks (CNNs) pertama, yang dapat mengenali pola visual.
* **Recurrent Neural Networks**, John Hopfield memperkenalkan jaringan Hopfield, yang merupakan implementasi awal dari jaringan saraf berulang (RNNs). Jaringan saraf berulang adalah algoritma revolusioner yang bekerja paling baik untuk data sekuensial.
* **Kemampuan Deep Learning,** Pada tahun 1986, Terry Sejnowski mengembangkan NETtalk, sistem text-to-speech berbasis jaringan saraf yang dapat mengucapkan teks bahasa Inggris.
* **Vanishing Gradient Problem,** Sepp Hochreiter menemukan dan membuktikan masalah gradien menghilang, yang memperlambat proses pembelajaran mendalam dan membuatnya tidak praktis.
* **GPU for Deep Learning**, Pada tahun 2009, Andrew Ng, Rajat Raina, dan Anand Madhavan, dengan makalah mereka “Large-Scale Deep Unsupervised Learning Using Graphics Processors,” merekomendasikan penggunaan GPU untuk deep learning karena jumlah core yang ditemukan di GPU adalah lebih banyak daripada yang ada di CPU.
* **ImageNet and AlexNet,** Pada tahun 2009, Fei-Fei Li meluncurkan database dengan 14 juta gambar berlabel, yang disebut ImageNet. Pada tahun 2012, Alex Krizhevsky merancang CNN yang dilatih GPU, AlexNet, yang meningkatkan akurasi model sebesar 75% dibandingkan model sebelumnya
* **Generative Adversarial Networks,** Pada tahun 2014, Ian Goodfellow datang dengan ide model jaringan saraf baru sekarang dikenal sebagai generative adversarial neural networks (GANs)
* **Power of Reinforcement Learning**, Pada tahun 2016, DeepMind melatih model pembelajaran penguatan mendalam, AlphaGo, yang dapat memainkan game GO
* **Turing Award to the Pioneers of Deep Learning,** Pada tahun 2019, tiga pionir AI, Yann LeCun, Geoffrey Hinton, dan Yoshua Bengio, berbagi penghargaan Turing. Penghargaan ini merupakan bukti yang menunjukkan pentingnya Deep Learning bagi komunitas ilmu komputer.

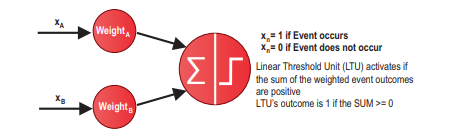
**Structure of Artificial Neural Networks**

* **McCulloch-Pitts Neuron**

****

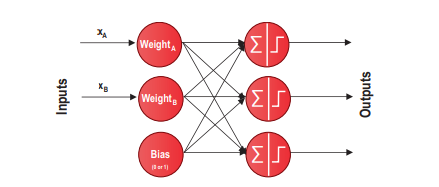
McCulloch Pitts Neuron diperkenalkan pada tahun 1943, dan hanya mampu melakukan operasi matematika dasar. Setiap peristiwa diberi nilai Boolean (0 atau 1), dan jika jumlah hasil peristiwa (0s dan 1s) melampaui ambang batas, maka neuron buatan akan menyala.

* **Linear Threshold Unit (LTU)**

****

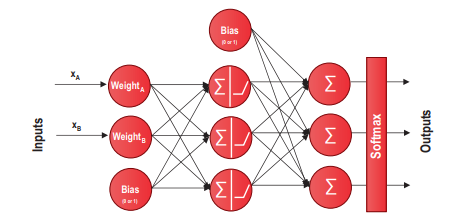
Dalam LTU, bobot ditetapkan untuk setiap peristiwa, dan bobot ini bisa negatif atau positif. Hasil setiap genap tetap diberi nilai Boolean (0 atau 1), tetapi kemudian dikalikan dengan bobot yang ditetapkan. LTU hanya diaktifkan jika jumlah hasil peristiwa tertimbang ini positif.

* **Perceptron**

****

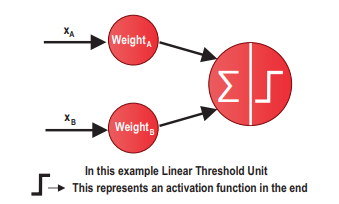
Algoritme perceptron dapat menyesuaikan bobot untuk memperbaiki perilaku jaringan saraf yang terlatih. Selain itu, istilah bias dapat ditambahkan untuk meningkatkan kinerja akurasi jaringan. Ketika hanya ada satu lapisan perceptron, itu disebut perceptron lapisan tunggal. Ada satu lapisan untuk output bersama dengan satu lapisan input yang menerima input. Ketika lapisan tersembunyi ditambahkan ke perceptron single-layer, kita berakhir dengan multilayer perceptron (MLP).

* **A Modern Deep Neural Network**

****

Fungsi aktivasi yang lebih kompleks daripada fungsi langkah (0 atau 1) seperti ReLU, Sigmoid, Tanh, dan Softmax, Jaringan saraf dalam modern biasanya memanfaatkan salah satu metode penurunan gradien untuk pengoptimalan

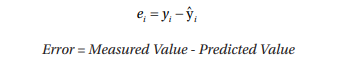
* **Activation Functions**

****

Fungsi aktivasi memperkenalkan langkah perhitungan akhir yang menambahkan kompleksitas tambahan ke jaringan saraf tiruan. Fungsi aktivasi meningkatkan kemampuan jaringan saraf untuk menggunakan informasi yang relevan dan menekan titik data yang tidak relevan. Ada sejumlah fungsi aktivasi berbeda yang dapat kita gunakan di jaringan saraf kita, seperti : Binary Step,Linear,Sigmoid (Logistic Activation Function),Tanh (Hyperbolic Tangent), ReLU (Rectified Linear Unit),Softmax, Leaky ReLU, Parameterized ReLU ,Exponential Linear Unit, Swish

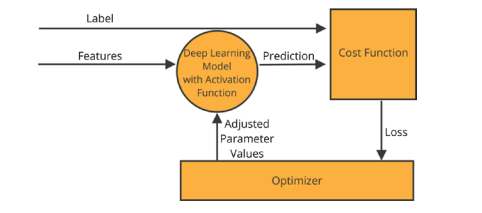
* **Loss (Cost or Error) Functions**

Fungsi loss adalah fungsi yang digunakan untuk mengukur kinerja model deep learning untuk data yang diberikan. Biasanya didasarkan pada istilah kesalahan, yang dihitung sebagai jarak antara nilai nyata (terukur) dan prediksi model yang dilatih.



Beberapa fungsi kerugian digunakan untuk evaluasi kinerja, dan memilih fungsi yang tepat merupakan bagian integral dari pembuatan model. fungsi kerugian juga dapat digunakan untuk penghargaan dalam pembelajaran penguatan. Root mean squared error (RMSE), mean squared error (MSE), mean absolute error (MAE), dan mean absolute persentase error (MAPE) adalah beberapa fungsi kerugian yang sesuai untuk masalah regresi. Untuk masalah klasifikasi biner dan multi kelas, kita dapat menggunakan variasi fungsi crossentropy (yaitu, logaritma).

* **Optimization in Deep Learning**

****

Fungsi optimasi membuat perubahan kecil pada bobot dan bias dan mengukur efek dari perubahan ini dengan fungsi kerugian. Proses ini membantu menemukan bobot optimal dan nilai bias untuk meminimalkan kesalahan dan memaksimalkan akurasi model. Ada beberapa algoritma optimasi dan tantangan yang dihadapi selama proses optimasi

* **Backpropagation**

Algoritma backpropagation adalah komponen penting dalam arsitektur jaringan saraf yang digunakan untuk iterasi secara paralel dengan pengoptimal. Ini berfungsi sebagai mekanisme sentral di mana jaringan saraf belajar.

* **Optimization Algorithms**

Algoritma optimasi dapat didefinisikan sebagai algoritma yang membantu algoritma lain untuk memaksimalkan kinerjanya tanpa penundaan. Algoritme pengoptimalan yang paling umum digunakan dalam tugas pembelajaran mendalam adalah : Adam,Stochastic gradient descent (SGD),Adadelta,Rmsprop,Adamax,Adagrad, dan Nadam.

* **Optimization Challenges**

Ada tiga tantangan optimasi yang sering kita temui dalam deep learning. Tantangan tersebut adalah (i) minima lokal, (ii) titik pelana, dan (iii) gradien hilang.

Local minima, Dalam pelatihan jaringan saraf, plot penurunan berat badan sederhana dengan minimum tunggal mungkin berguna untuk memvisualisasikan hubungan antara berat dan kerugian yang dihitung untuk tujuan pendidikan. Titik pelana -Titik pelana adalah titik stabil dalam grafik yang tidak dapat ditentukan oleh algoritme apakah itu minimum lokal atau maksimum lokal. Gradien yang Menghilang, Solusi efektif untuk masalah gradien hilang adalah dengan menggunakan ReLU sebagai fungsi aktivasi di lapisan tersembunyi.

**Overfitting and Regularization**

**Overfitting**

Solusi untuk masalah underfitting adalah membangun model yang baik dengan fitur yang berarti, memasukkan data yang cukup, dan pelatihan yang cukup. Di sisi lain, lebih banyak data, menghapus fitur yang berlebihan, dan validasi silang adalah metode yang tepat untuk mengatasi masalah overfitting.

**Regularization**

Regularisasi adalah teknik untuk melawan overfitting. Ada sejumlah metode yaitu : Early stopping, Dropout, L1 and L2 regularization, Data augmentation. Early stopping : adalah strategi yang sangat sederhana – namun efektif – untuk mencegah overfitting. Menetapkan jumlah epoch yang cukup (langkah-langkah pelatihan) sangat penting untuk mencapai tingkat akurasi yang baik, Dropout : Dengan mengaktifkan dropout, model kami untuk sementara menghapus beberapa neuron atau lapisan dari jaringan, yang menambahkan noise tambahan ke jaringan saraf, L1 and L2 regularization : . Regularisasi L1 dan L2 sangat membantu ketika berhadapan dengan sekumpulan besar fitur, Augmentasi : Augmentasi data adalah metode untuk menambah jumlah data latih.

**Feature Scaling**

Penskalaan fitur adalah metode untuk menormalkan rentang fitur sehingga jaringan saraf dapat bekerja lebih akurat. Ada beberapa metode penskalaan fitur yang tercantum sebagai berikut:

1. Standardisasi: Ini menyesuaikan nilai setiap fitur untuk memiliki rata-rata nol dan varians unit.
2. Normalisasi Min-Maksimum (Penskalaan Ulang): Ini menskalakan nilai setiap fitur antara [0, 1] dan [-1, 1].
3. Etc

**Final Evaluations**

Dalam bab ini, kami membahas garis waktu jaringan saraf tiruan dan pembelajaran mendalam. Ini membantu kami untuk memahami bagaimana konsep yang kami gunakan dalam kehidupan profesional kami menjadi hidup setelah bertahun-tahun penelitian. Kabar baiknya adalah berkat TensorFlow, kami dapat menambahkan komponen ini ke jaringan saraf kami dalam hitungan detik.