

# Pendahuluan

Pada tutorial ini saya akan membahas tentang aspek praktis dari algoritma jaringan saraf tiruan menggunakan pustaka NeuroLab yang berbasis pada bahasa pemrograman Python. Jaringan saraf tiruan saat ini sedang menjadi buah bibir dalam perkembangan teknologi dunia semenjak Alex Krizhevsky (pendiri laboratorium Dossa), Ilya Sutskever (ilmuwan kepala di laboratorium OpenAI), dan Geoffrey E. Hinton (guru besar di Departemen Ilmu Komputer, Universitas Toronto) menerbitkan makalah tentang penerapan pembelajaran mendalam (*deep learning*) untuk mengklasifikasikan 1,2 juta citra menggunakan arsitektur yang nantinya dikenal sebagai AlexNet (Krizhevsky dkk., 2012) pada ajang bergengsi *Conference and Workshop on Neural Information Processing Systems* (NeurIPS) yang diselenggarakan di Stateline, Nevada, Amerika Serikat tahun 2012 silam. Semenjak itu, hampir setiap perusahaan teknologi raksasa, seperti Google; Facebook; Baidu; dll melakukan ekspansi besar - besaran untuk mendanai tim riset algoritma ini.

Algoritma jaringan saraf tiruan saat ini merupakan bagian penting dari riset dan pengembangan teknologi buatan yang menjadi sendi utama dalam revolusi industri ke-4, seperti yang diterangkan oleh ekonom dunia Karl Schwab (Schwab, 2016). Presiden kita, Ir. H. Joko Widodo (beserta staf - staf kepresidenan-nya), yang menurut hemat saya juga tidak mengerti apa - apa tentang algoritma jaringan saraf tiruan, sempat mewacanakan perampingan struktur jabatan aparatur sipil negara (ASN) dengan memanfaatkan dengan menggunakan aplikasi kecerdasan buatan (Tempo, 2019), yang saya anggap sebagai suatu utopia kosong yang umum digaungkan orang - orang bodoh yang belum belajar sulitnya penerapan algoritma jaringan saraf tiruan di dunia nyata.

Daripada membicarakan politik yang bukan level-nya saya, maka baiknya kita kembali ke isi buku singkat ini, kita tidak akan membahas hal - hal yang tinggi - tinggi juga muluk - muluk seperti penerapan *Generative Adversarial Network* (GAN) (Goodfellow dkk., 2014), untuk membuat foto *bohongan* seperti yang saat ini menjadi kontroversi. Kita akan membahas hal - hal sederhana, seperti:

- konsep - konsep dasar dalam algoritma jaringan saraf tiruan,
- penerapan klasifikasi menggunakan perseptron tunggal,
- penerapan algoritma jaringan saraf buatan berlapis tunggal (*single-layer neural network*),
- penerapan algoritma jaringan saraf buatan berlapis banyak (*multi-layer neural networks*), dan
- pengantar analisis data sikuensial menggunakan algoritma *Recurrent Neural Networks* (RNNs)

Selain itu, saya juga sebetulnya telah menyiapkan *script* Python untuk melakukan pengenalan karakter optis (menggunakan [dataset tulisan tangan](#) yang awalnya disiapkan oleh Rob Kassel (MIT Spoken Language Systems Group) namun telah dibersihkan, dinormalisasi, dan dirasterisasi oleh Ben Taskar, saat ini menjabat sebagai lektor kepala di Departemen Ilmu Komputer dan Informatika di Universitas Pennsylvania, ketika yang bersangkutan masih menjadi mahasiswa doktoral di Stanford di bawah supervisi Daphne Koller., namun mungkin karena saya yang kurang kompeten atau *bodohnya* pembuat pustaka, atau karena *laptop* saya yang sudah terlampau renta, maka *script* tersebut menghasilkan *MemoryError*. Untuk pembaca yang tertarik untuk membantu atau *mengata-ngatain* saya dipersilahkan untuk melihat[nya](#). Seluruh materi yang saya sampaikan diterapkan pada pustaka [NeuroLab versi 0.3.5](#) (alih - alih pustaka *deep learning* Python lainnya yang lebih populer seperti: TensorFlow (Abadi dkk., 2016), Keras (Chollet dkk., 2015), dan PyTorch (Paszke dkk., 2019)), yang dikembangkan oleh tiga orang pengembang Rusia dengan nama samaran: maksimovVva; zueve; dan solarjoe, untuk mempermudah para akademisi Indonesia di luar bidang ilmu komputer yang lebih terbiasa dengan *Neural Network Toolbox* (NNT) yang dijalankan pada piranti lunak berbayar: MATLAB<sup>®</sup>. Saya juga pada dasarnya akan mengikuti struktur buku Prateek Joshi (Joshi, 2017) dalam alur pembagian materi di tutorial ini. Oh, ya satu hal lagi yang perlu saya ingatkan karena tutorial ini bersifat praktis, saya tidak akan membahas tentang konsep matematis algoritma ini. Disarankan pembaca menunggu buku yang saya dan Dr. Dasapta Erwin

Irawan (Teknik Geologi ITB) akan siapkan pada pertengahan tahun depan (jika saya masih hidup dan sehat) untuk pembahasan lebih mendasar tentang konsep *back-propagation*, fungsi aktivasi, dll. Atau kalau sudah tidak sabar, *monggo* untuk melihat buku - buku seperti: Goodfellow, dkk. (2016), Burkov (2019), dan (mungkin yang lebih *asik* dan praktis bagi saya pribadi yang *tolol* dalam matematika) Géron (2019). Tutorial ini juga tidak membahas implementasi *deep learning* menggunakan GPU seperti NVIDIA CUDA dan OpenCL, oleh karena itu pembaca diharapkan mencari tahu hal tersebut secara mandiri di mesin pencariinya masing - masing. Sebagai tambahan lagi, di sini saya juga mengharapkan pembaca punya sedikit pengalaman dengan bahasa pemrograman Python sampai tingkatan pemrograman *ber-gaya* fungsional dan mahir dalam penggunaan pustaka NumPy (Oliphant, 2006) serta matplotlib (Hunter, 2007) karena saya tidak akan membahas lagi apa itu vektorisasi *array*, apa itu `if __name__ == '__main__':`, dan tata cara visualisasi sederhana menggunakan modul pyplot di pustaka matplotlib. Terakhir, bagi pembaca yang tertarik untuk memperoleh bacaan yang saya sertakan di daftar pustaka secara ilegal dan tidak mau kena pasal, saya siapkan semuanya secara cuma - cuma di laman [GitHub saya](#).

Untuk melakukan instalasi ikuti prosedur sebagai berikut:

1. Kunjungi <https://docs.conda.io/projects/conda/en/latest/user-guide/install/> untuk instalasi Miniconda versi 3.
2. Ikuti prosedur instalasi (jangan lupa atur PATH -nya).
3. Ikuti prosedur instalasi git pada situs:  
<https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>.
4. Buku *Command Line Interface* (CLI), jalankan perintah:

```
git clone git@github.com:sandyherho/NeuroLab-tutor.git
```

5. Lakukan instalasi lingkungan virtual conda dengan menjalankan perintah di folder hasil *cloning* tersebut:

```
conda env create -f environment.yml
```

6. Aktifkan lingkungan virtual dengan perintah:

```
conda activate neuroLabs
```

7. Untuk mengaktifkan Jupyter Notebook, jalankan perintah:

```
jupyter notebook
```

8. Jika sudah terbuka di *browser* masing - masing sesi Python interaktif dapat dimulai.
9. Untuk mengakhiri sesi tekan tombol <CTRL> + C di CLI, dan jalankan perintah sebagai berikut untuk menonaktifkan lingkungan virtual conda:

```
conda deactivate
```