

**Spesifikasi Tugas Besar**  
**IF3270 Pembelajaran Mesin**  
**Teknik Informatika STEI ITB**  
**Semester 2 2022/2023**

Tugas besar ini dikerjakan berkelompok, terdiri atas 4 mahasiswa (boleh gabungan mahasiswa K1, K2, dan K3). Terdapat 2 bagian yang akan dikerjakan, dengan deadline yang berbeda.

Catatan:

1. Tugas dikumpulkan berupa hasil download notebook dalam dua format yaitu file .ipynb dan pdf.
2. Hanya salah satu anggota kelompok saja yang mengumpulkan file tugas pada classroom kuliah ini. Penamaan file yang dikumpulkan: TubesX\_[NIM salah satu anggota].zip.  $X \in \{A,B,C\}$
3. Pengumpulan yang terlambat tidak diperbolehkan, perhatikan batas akhir pengumpulan setiap bagian (waktu situs kuliah ini).

**Bagian A: Implementasi Forward Propagation untuk Feed Forward Neural Network**

Dikumpulkan: Jumat, 7 April 2023 jam 23.59

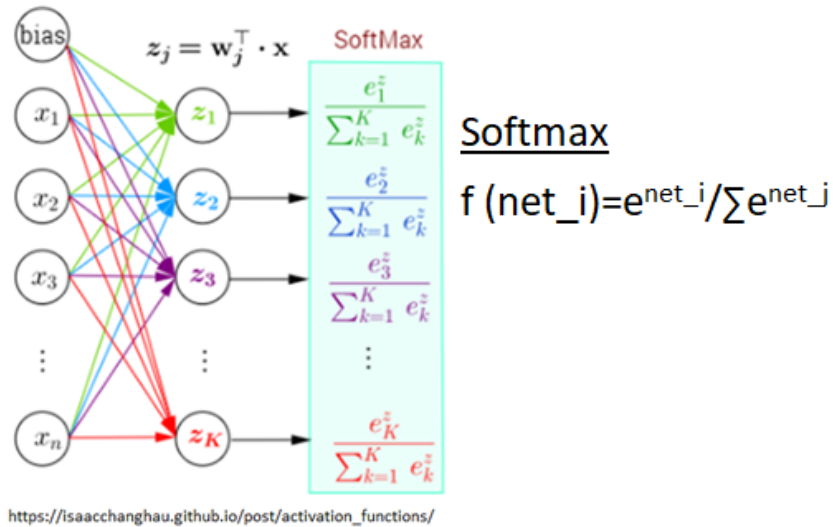
1. Rancanglah struktur file teks eksternal yang akan dipakai untuk menyimpan model dalam bentuk struktur jaringan (jumlah layer, fungsi aktivasi dan jumlah neuron per layer) dan bobot dari FFNN. Setiap layer dibatasi memiliki neuron dengan fungsi aktivasi yang sama, sedangkan antar layer diperbolehkan memiliki neuron dgn fungsi aktivasi yang berbeda. Fungsi aktivasi yang dikenali adalah linear, sigmoid, ReLU, dan softmax.

linear:  $f(\text{net}) = \text{net}$

ReLU:  $f(\text{net}) = \max\{0, \text{net}\}$

sigmoid:  $f(\text{net}) = 1/(1 + e^{-(\text{net})})$

Softmax:  $f(\text{net}_i) = e^{(\text{net}_i)} / \sum e^{(\text{net}_i)}$



2. Implementasi fitur load (membaca) model FFNN dari file teks.
3. Implementasi forward propagation untuk FFNN dengan kemampuan:
  - a. Menampilkan model berupa struktur jaringan dan bobotnya, formatnya bebas.
  - b. memprediksi output untuk input 1 instance.
  - c. Memprediksi output untuk input batch sejumlah instances.
4. Pengujian kebenaran fungsional dari forward propagation yang diimplementasikan dilakukan dengan menggunakan kasus uji yang disiapkan oleh asisten.
5. Deliverables: a) source code, b) laporan berisi penjelasan implementasi, hasil pengujian, perbandingan dengan hasil perhitungan manual, dan pembagian tugas setiap anggota kelompok.

## Bagian B: Implementasi Mini-batch Gradient Descent

Dikumpulkan: Jumat, 5 Mei 2023 jam 23.59

1. Implementasi backpropagation dengan mini-batch gradient descent sesuai materi kuliah.
  - a. Pada mini-batch gradient descent, update bobot dilakukan per mini-batch. Satu train data dibagi menjadi beberapa mini-batch sesuai parameter **batch\_size**.
  - b. Setiap layer dibatasi memiliki neuron dengan fungsi aktivasi yang sama, sedangkan antar layer diperbolehkan memiliki neuron dengan fungsi aktivasi yang berbeda.
  - c. Algoritma yang diimplementasikan adalah backpropagation dengan fungsi **aktivasi** linear, sigmoid, ReLU, dan softmax.

- d. Untuk linear, sigmoid, dan ReLU, gunakan fungsi loss berupa sum of

$$E = \frac{1}{2} \sum_{k \in \text{outputs}} (t_k - o_k)^2$$

squared errors:

Untuk softmax, gunakan fungsi loss berupa cross entropy:

$$E = -\log(p_k), k=\text{target}$$

- e. Turunan dari fungsi aktivasi:

Linear:  $f(x)=x \rightarrow df/dx=1$ .

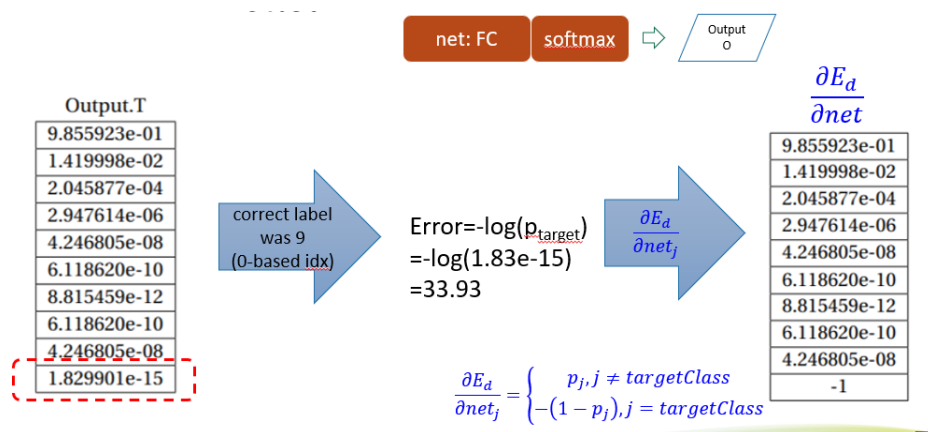
$$\frac{d}{dx} \text{relu}(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases}$$

$$\frac{d}{dx} \text{sigmoid}(x) = \text{sigmoid}(x)(1 - \text{sigmoid}(x))$$

Softmax:

$$\frac{\partial E_d}{\partial \text{net}_j} = \begin{cases} p_j, j \neq \text{targetClass} \\ -(1 - p_j), j = \text{targetClass} \end{cases}$$


Contoh perhitungan turunan softmax:



- f. Dalam mengimplementasikan backpropagation, implementasi berupa perhitungan gradient dengan hasil aturan rantai sebelum mengupdate bobot dengan cara mengalikan gradient\*-1\***learning rate**. Perhatikan aturan rantai perhitungan gradient utk update bobot ke output layer,

berbeda dengan hidden layer. Untuk update bobot output layer  $w$  yang menerima input  $h$ , gradient dihitung sebagai  $dE/dw = dE/net \cdot dnet/dw$ . Untuk aktivasi output neuron selain softmax, diimplementasikan  $dE/dw = dE/dOut \cdot dOut/dNet \cdot dNet/dw$  sehingga kita mengalikan hasil dari 3 suku yaitu  $-(t-out)$ ,  $out(1-out)$ , dan  $x$ . Pada softmax, sudah diberikan langsung  $dE/net$  sehingga kita mengalikan  $dE/dw = dE/dNet \cdot dNet/dw$ .

Chain rule to compute gradient:



$$\frac{\partial E_d}{\partial w_{ji}} = \frac{\partial E_d}{\partial o_j} \frac{\partial o_j}{\partial net_j} \frac{\partial net_j}{\partial w_{ji}} = -(t_j - o_j) o_j (1 - o_j) x_{ji}$$

$$\frac{\partial net_j}{\partial w_{ji}} = x_{ji}; net_j = \sum_{i \in [0..n_j]} x_{ji} w_{ji}$$

$$\frac{\partial o_j}{\partial net_j} = \frac{\partial \sigma(net_j)}{\partial net_j} = o_j (1 - o_j)$$

$$\frac{\partial E_d}{\partial o_j} = \frac{\partial}{\partial o_j} \frac{1}{2} \sum_{k \in outputs} (t_k - o_k)^2 = \frac{1}{2} \cdot 2 \cdot (t_j - o_j) \cdot -1 = -(t_j - o_j)$$

- g. Kondisi berhentinya adalah error kumulatif  $\leq$  **error threshold** atau **maksimum iterasi** tercapai. Error threshold dan maksimum iterasi merupakan parameter dari mini-batch gradient descent.
  - h. Definisikanlah parameter apa saja yang bisa ditangani dalam implementasi kelompok Anda. Parameter yang wajib ada adalah: struktur jaringan (jumlah layer, jumlah neuron setiap layer, fungsi aktivasi setiap layer), learning-rate, error threshold, max\_iter, batch\_size.
  - i. Pengujian kebenaran fungsional dari backprop yang diimplementasikan dilakukan dengan menggunakan kasus uji yang disiapkan oleh asisten.
  - j. Lakukan pembelajaran backpropagation dengan hasil implementasi ini untuk dataset iris untuk semua data (full training), dan menampilkan modelnya.
2. Implementasi penyimpanan model hasil pembelajaran, lalu menguji hasilnya dengan menggunakan fungsi load model yang sudah diimplementasikan pada bagian sebelumnya.
  3. Deliverables: a) source code, b) laporan berisi penjelasan implementasi, hasil eksekusi (langkah 2), perbandingan dengan hasil MLP sklearn, dan pembagian tugas setiap anggota kelompok.