

**Spesifikasi Tugas Besar**  
**IF3270 Pembelajaran Mesin**  
**Teknik Informatika STEI ITB**  
**Semester 2 2020/2021**

Tugas besar 1 ini dikerjakan berkelompok, terdiri atas 4 mahasiswa (boleh gabungan mahasiswa K1, K2, dan K3). Terdapat 3 bagian yang akan dikerjakan, dengan deadline yang berbeda.

Catatan:

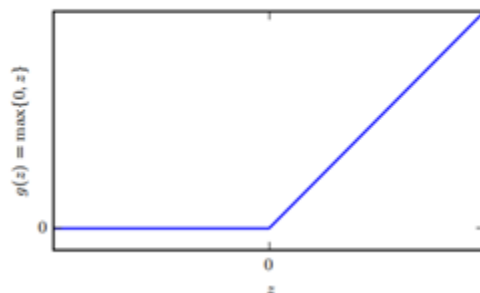
1. Tugas dikumpulkan berupa hasil download notebook dalam dua format yaitu file .ipynb dan pdf.
2. Hanya salah satu anggota kelompok saja yang mengumpulkan file tugas pada classroom kuliah ini. Penamaan file yang dikumpulkan: TubesX\_[NIM salah satu anggota].zip.  $X \in \{A,B,C\}$
3. Pengumpulan yang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

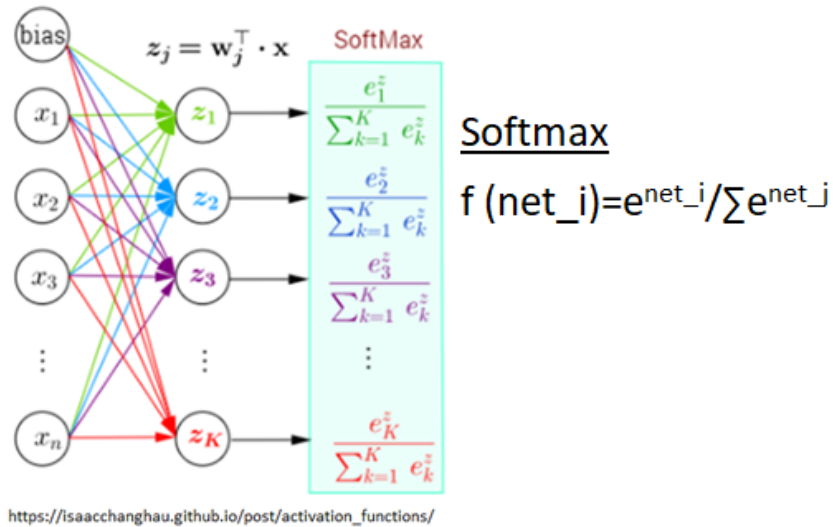
**Bagian A: Implementasi Forward Propagation untuk Feed Forward Neural Network**

Dikumpulkan: Jumat, 5 Maret 2021 jam 23.55

1. Implementasi forward propagation untuk FFNN dengan kemampuan:
  - a. membaca model FFNN (struktur dan bobot). Rancanglah terlebih dahulu file eksternal yang akan dipakai untuk menyimpan model FFNN. Fungsi aktivasi yang dikenali adalah linear, sigmoid, ReLU, dan softmax. Setiap layer dibatasi memiliki neuron dgn fungsi aktivasi yang sama, sedangkan antar layer diperbolehkan memiliki neuron dgn fungsi aktivasi yang berbeda.

$$\text{ReLU}(z) = \max\{0, z\}$$





- b. Menampilkan model berupa struktur dan koefisiennya, formatnya bebas.
  - c. memprediksi output untuk input 1 instance. Pengujian dilakukan dengan kedua model XOR dari slide kuliah.
  - d. Memprediksi output untuk input batch sejumlah instances. Pengujian dilakukan dengan kedua model XOR dari slide kuliah.
2. 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: Sabtu, 27 Maret 2021 jam 23.55

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

squared errors:

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

Untuk sigmoid, gunakan fungsi loss berupa cross entropy:

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

e. Turunan dari fungsi aktivasi:

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

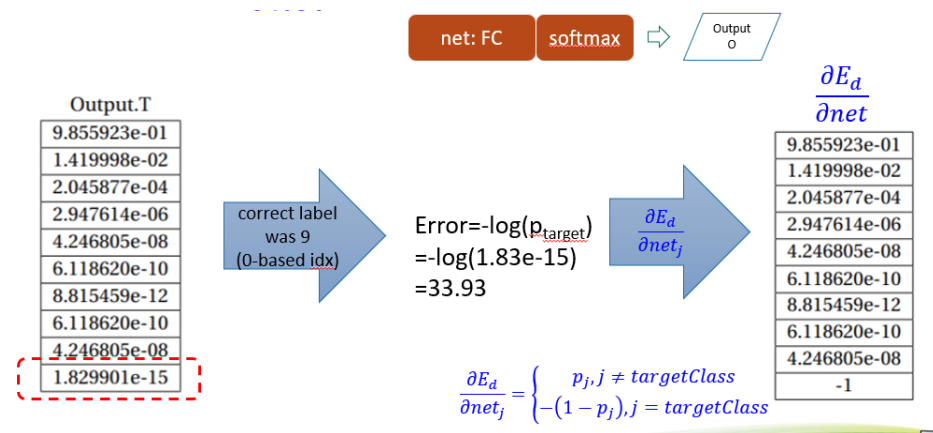
$$\frac{d}{dx} \text{relu}(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases}$$

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

Softmax:

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

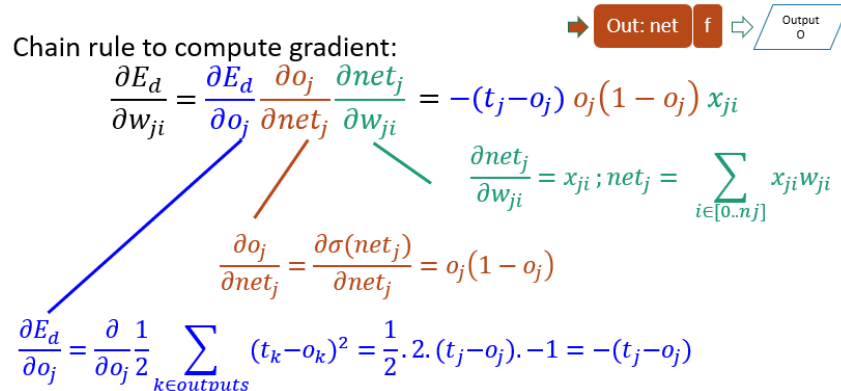
Contoh perhitungan turunan softmax:



f. Dalam mengimplementasikan backpropagation, implementasi berupa perhitungan gradient dengan hasil aturan rantai sebelum mengupdate bobot dengan cara mengalikan  $\text{gradient} \cdot -1 \cdot \text{learning rate}$ . 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  $dE/dw = dE/\text{net} \cdot d\text{net}/dw$ . Untuk aktivasi output neuron selain softmax, diimplementasikan  $dE/dw = dE/d\text{Out} \cdot d\text{Out}/d\text{Net} \cdot d\text{Net}/dw$  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  $dE/dw = dE/dNet * dNet/dw$ .

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 (TBD)
  - j. Lakukan pembelajaran backpropagation dengan hasil implementasi ini untuk dataset iris untuk semua data (full training), dan menampilkan modelnya.
2. Deliverables: a) source code, b) laporan berisi penjelasan implementasi, hasil eksekusi (langkah 2), perbandingan dengan hasil MLP sklearn, dan pembagian tugas setiap anggota kelompok.

### Bagian C: Persiapan Eksperimen

Dikumpulkan: Sabtu, 3 April 2021 jam 23.55

Dengan menggunakan jupyter notebook, gunakanlah hasil implementasi kelompok masing-masing.

1. Implementasi pembentukan confusion matrix saat prediksi batch, dan implementasi juga perhitungan akurasi, precision, recall, dan F1.
2. Lakukan pengujian dengan membandingkan confusion matrix dan perhitungan kinerja dari sklearn.
3. Lakukan pembelajaran FFNN untuk dataset iris dengan skema split train 90% dan test 10%, dan menampilkan kinerja serta confusion matrixnya.

4. Lakukan pembelajaran FFNN untuk dataset iris dengan skema 10-fold cross validation, dan menampilkan kinerjanya.
5. Simpan (save) model/hipotesis hasil pembelajaran skema full training ke file eksternal.
6. Baca (load) model/hipotesis dari file eksternal
7. Buat instance baru dengan memberi nilai untuk setiap atribut, lalu prediksi dengan memanfaatkan model/hipotesis dari hasil 6.
8. Lakukan analisis hasil dari 2 dan 3.